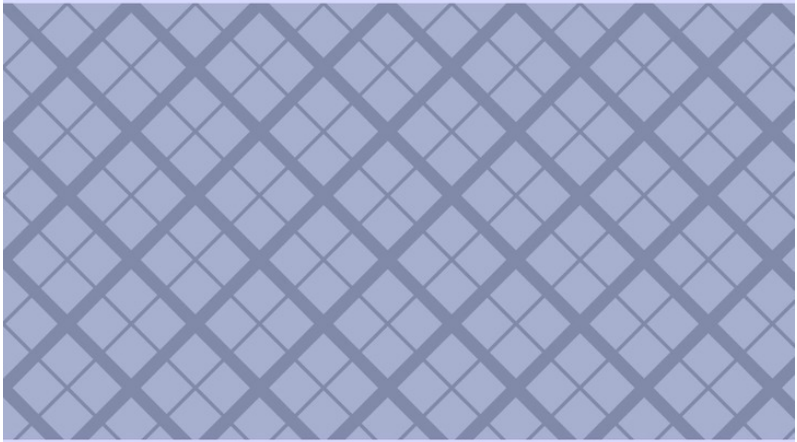
Иван Андреевич Трещев
Владимир Александрович Тихомиров



# ПРОГРАММИРОВАНИЕ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

Android и WP. Учебный курс

**Иван Андреевич Трещев**
**Владимир Александрович Тихомиров**

# Программирование для мобильных платформ. Android и WP. Учебный курс

*http://www.litres.ru/pages/biblio_book/?art=55732126*
*ISBN 9785449894588*

## Аннотация

Данная книга содержит основные материалы курсов программирования для мобильных платформ, используемых на кафедрах «Математическое обеспечение и применение ЭВМ» и «Информационная безопасность автоматизированных систем» ФГБОУ ВО КнАГУ.

# Содержание

# Программирование для мобильных платформ Android и WP. Учебный курс

**Иван Андреевич Трещев**
**Владимир Александрович Тихомиров**

*Общий анализ* Анастасия Сергеевна Ватолина



ISBN 978-5-4498-9458-8
Создано в интеллектуальной издательской системе Ridero

# Раздел 1. Windows Phone – Silverlight и xna

## ВВЕДЕНИЕ

Много чего можно отнести к мобильным программируемым устройствам. В принципе – это любая программируемая компьютерная техника, которую человек может перетащить с собой в дипломате. В наш курс будут входить только:

– сотовые телефоны;
– смартфоны;
– коммуникаторы;
– КПК (карманные персональные компьютеры).

## Сотовые телефоны

Сотовый телефон оснащен прошивкой – выполняющей роль примитивной операционной системы (ОС), компоненты которой простой смертный, не знающий языка программирования, вряд ли сможет изменить. Однако большинство современных аппаратов помимо неизменяемой прошивки имеют в своем арсенале программную платформу Java2ME (Java 2 Micro Edition), которая позволяет закачивать на те-

лефон приложения, написанные на языке Java, в том числе и игры. Скорее всего, по мере удешевления смартфонов сотовые телефоны, а вместе с ними и платформа Java2ME, канут в лету. Однако на данный момент в интернете можно найти довольно большое количество приложений, которые для неё подходят. Кроме этой платформы, существуют ещё две: **Mophun** (живет только в старых телефонах **Sony Ericsson**, например **SonyEricssonT610)** и **BREW.** Последняя увидела свет позже, чем Java2ME, в 2001 году, и изначально предназначалась для CDMA-телефонов, затем была адаптирована и для телефонов стандарта GSM, но широкого распространения пока так и не получила, по крайней мере в России.

## Смартфоны

Название «смартфонов» произошло от двух английских слов: «Smart», что в переводе означает «умный», и «Phone» – телефон. То есть смартфон – это умный телефон. А какой телефон можно считать «умным»? Разумеется, клиента электронной почты, WAP-браузера и «продвинутого» редактора рингтонов тут недостаточно. Речь идет о других, ***компьютерных*** функциях аппарата. То есть у него должна быть настоящая операционная система, большой дисплей, Bluetooth и/или инфракрасный порт. Также «ум» телефона подчеркивают возможность синхронизации с ПК, наличие слотов рас-

ширения и достаточный объем памяти для установки приложений.


Рисунок 1. Разновидности смартфонов

От стоящих ниже на лестнице эволюции телефонов смартфонам достались телефонная клавиатура и отсутствие сенсорного дисплея.

## Коммуникатор

Коммуникатор – это карманный компьютер с функциями телефона. Это значит, что в устройстве зачастую не бывает телефонной клавиатуры (рис.), в обязательном порядке имеются полноценная операционная система и сенсорный экран. Многие модели коммуникаторов делаются очень просто – берется серийная модель смартфона, добавляется

GSM-модуль и выводится на рынок в качестве новой модели с ценой сотни на полторы больше, чем у «родителя».


Рисунок 2. Разновидности коммуникаторов

От этого, правда, часто страдает эргономика, которая у большинства КПК и так не блещет. Либо получается начисто лишенная всех удобств, но крайне функциональная «коробка», либо непроизводительная и нефункциональная, но очень красивая поделка. «Золотая середина» все же иногда встречается. Общепризнанным эталоном коммуникатора является Sony Ericsson P900/910 (рис. 2). Но за удобство приходится платить.

Удобству использования коммуникаторов сильно способствуют разные аксессуары, количество которых намного больше, чем у смартфонов. При помощи двух-трех дополнительных аксессуаров можно превратить коммуникатор в ма-

шину, ничем по удобству не уступающую ноутбуку. Тем, кому необходимо писать большие тексты или заполнять огромные отчеты, наверняка поможет дополнительная клавиатура. Есть даже резиновые клавиатуры, которые при желании можно свернуть в трубочку. А если созданные на коммуникаторе работы захочется напечатать, минуя настольный компьютер – тоже нет проблем. Благо для мобильных устройств выпущено немало компактных и удобных принтеров. Так что владелец коммуникатора вполне может начать работать, едва выйдя из дома. Что, как известно, никогда и никем не возбраняется.

Рисунок 3. Разновидности коммуникаторов с клавиатурами

## Карманный персональный компьютер

КПК – это карманные персональные компьютеры со своей

памятью, процессором, слотами расширения, звуковой системой, ну и, конечно же, дисплеем. КПК от своих старших собратьев отличаются меньшим дисплеем, который, к тому же, реагирует на прикосновение специальной палочки – стилуса – и flash-памятью, которая, в отличие от винчестера, занимает меньше физического места.

Чаще всего КПК сравнивают по функциональности с ноутбуками. Хотя это совершенно разные устройства, предназначенные для использования в разных целях. Основное различие их в том, что для работы с ноутбуком Вам, прежде всего, нужно где-то его поставить и самому принять удобное положение, а также дождаться загрузки операционной системы. С КПК все гораздо проще. В метро, в трамвае, сидя или стоя, Вы запросто включаете его и, управляя стилусом либо цифровой клавиатурой (если есть), будете спокойно работать. Процесс включения занимает считанные секунды.

Также немаловажно то, что при работе с ноутбуком Вы пользуетесь клавиатурой для набора текстов, а в КПК для этого используется стилус. Для набора текста используется «виртуальная клавиатура». Также можно писать слова от руки, как Вы это делаете ручкой на бумаге. С первого взгляда это не очень удобно, но, потренировавшись, можно вполне быстро набирать небольшие заметки или статьи. При желании, никто не мешает купить дополнительную клавиатуру, которая будет соединяться с Вашим КПК по Bluetooth или через USB-порт.

Раньше считалось, что КПК могут использоваться только для набора текстов, прослушивания музыки и простеньких игр – из-за слабых процессоров и малого наличия памяти. На сегодняшний день современные КПК обладают высокоскоростными процессорами по 400, 520 и выше МГц, что вполне достаточно для просмотра несжатого видео в формате DivX, 3D игр типа Doom, работы с базами данных и большими электронными таблицами. Память тоже можно расширить, купив дополнительную карту на 1, 2 или больше ГБ, благо такие карты дешевеют с каждым днем.

Рисунок 4. Разновидности карманных персональных компьютеров (КПК)

Рисунок 4. Разновидности карманных персональных компьютеров (КПК)

От смартфонов КПК отличаются только тем, что в них не вставляется SIM карта и в них нет телефонной связи. В противовес – аппаратные мощности КПК (процессор, память, размер экрана и т.д.) существенно выше.

# 1 Обзор операционных систем мобильных устройств

В сотовых телефонах, как говорилось выше, нет операционной системы, как таковой, там – прошивка, на которую «натянута» виртуальная Java машина, исполняющая программы, называемые «мидлетами».

Настоящие операционные системы начинаются со смартфонов. Смартфоны и коммуникаторы имеют возможность устанавливать дополнительное программное обеспечение, зачастую от сторонних разработчиков, для добавления новых возможностей и расширения функциональности.

В коммуникаторах и смартфонах широкое распространение получили операционные системы:

– Symbian OS
– Windows Mobile
– Palm OS
– iPhone OS
– BlackBerry OS
– Samsung Bada
– Системы на базе Linux:
Google Android,
Palm webOS,
Access Linux Platform,
Nokia Maemo.

Кроме ОС существуют еще достаточно интересные приложения, дополняющие саму операционную систему, расширяющие ее функциональность и меняющие внешний вид. Как пример, можно вспомнить TouchFLO 3D для коммуникаторов HTC или фирменный интерфейс TouchWIZ, используемый в мобильных устройствах Samsung.

## 1.1 Palm OS


Palm OS система – достаточно редкая. Из различных околокомпьютерных СМИ мы слышим, что у Palm «не все в порядке». И это не удивительно, учитывая, что сейчас не каждый сведущий в ОС человек даст однозначный ответ на вопрос: «Кому принадлежат права на Palm OS?»


Рисунок 1.1 Дизайн
Palm OS

Palm OS Garnet принадлежит ACCESS, но Palm Inc купила у ACCESS «пожизненное право» на исходный код Palm OS 5.4 Garnet, а это значит, что она имеет право разрабатывать свои продукты на этой основе. Также заявлена и шестая версия системы, но под ее парусами еще не работает ни одно устройство.

Несмотря на все проблемы, число «пальмоводов» более чем внушительно, а значит, эту операционку рано сбрасывать со счетов. Что большинству пользователей нужно от КПК? Максимальное использование дисплея, честная надежность, мультимедийность, безболезненная синхронизация с ПК, приличное время работы без подзарядки. Все это есть в устройствах на основе Palm OS. Плюсов много, а минусы…

Вообще разработчики сейчас стремятся вперед, множат плюсы, во многом забывая о минусах. Болезнь Palm OS еще с детства – это отсутствие нормальной многозадачности. Иными словами многозадачность здесь реализована по шаблону «почувствуй себя пользователем мобильного телефона», то есть, запустив одно приложение, Вы не сможете запустить параллельно другое. К тому же сложно положительно охарактеризовать такое собирательное понятие, как мультимедийность, говоря о его реализации его в Palm OS.

Достоинства:

– Нетребовательна к ресурсам;

– Очень удобный интерфейс пользователя;

– Удобная синхронизация с ПК;

– Надежность;

Недостатки:

– Отсутствует полноценная многозадачность;

– Не развиты мультимедийные функции;

– Система не развивается (хотя возможно компания HP сможет это преодолеть);

# 1.2 Symbian OS

До последнего времени, это самая распространенная операционная система для смартфонов. По прогнозу аналитиков из компании «Garnter», в 2012-м году Symbian все еще будет самой распространенной операционной системой для смартфонов, однако ее доля уменьшится с почти 50% до 39% (на фото – Samsung i8910 Omnia HD).

Symbian OS изначально создавалась исключительно для смартфонов, прототипом для нее послужила операционная система EPOC 32. Впервые эта операционка от компании Psion – одного из пионеров рынка КПК – была использована в КПК Psion Series 5 в 1997 году. Она была создана для работы с процессорами ARM, традиционными для мобильных устройств, и обеспечивала работу с клавиатуры и через сенсорный экран. Несомненным плюсом в пользу новой систе-

мы стало разграничение графического интерфейса и другого ПО – это позволило адаптировать систему под устройства с любыми характеристиками экрана и клавиатуры, а производителям – использовать разные интерфейсы. ПО EPOC 32 также отличалось компактностью, что позволяло бес проблем использовать его на устройствах с ограниченными ресурсами. EPOC32 Release 5u (Symbian OS v5) была оптимизирована для работы с коммуникационными протоколами (например, с протоколами сотовой связи). В ней появилась возможность работы с Интернетом и почтой, обработки файловых вложений и SMS. На ее основе работал популярный смартфон начала 2000-х – Ericsson R380.


Рисунок 1.2 Дизайн Symbian OS

Самыми распространенными платформами на базе

Symbian сейчас являются: Series 60 (самая популярная платформа, используется на моделях Nokia, Panasonic, Samsung, Lenovo, LG и Sony Ericsson и д.р.), Series 80 (использовалась в некоторых моделях Nokia), Series 90 (сейчас используется только на Nokia 7710), UIQ (модели Nokia, Benq, Motorola, Arima, Sony Erisson, и MOAP (закрытая платформа, устанавливается на телефонах Fujitsu, Sony Ericsson, Mitsubishi и Sharp).

Series 60 создавалась для смартфонов с телефонной клавиатурой, в ОС закладывалась поддержка экранов высокого разрешения, управление приложениями через сенсорный ввод.

Разработкой и продвижением данной операционки занимается некоммерческая организация Symbian Foundation, созданная в 1998 году, в ее состав входят 40 компаний, среди которых «Samsung», «Nokia», «LG Electronics», «Sharp», «Sony Ericsson», «Huawei», Motorola, Siemens, Panasonic, Fujitsu, Sony, Sanyo, Ericsson, AT&T, Psion, STMicroelectronics, Texas Instruments и другие.

Программ, предназначенных для Symbian OS, существует огромное количество; их можно узнать по расширению SIS. Файлы в формате SIS представляют собой самораспаковывающийся архив. Работоспособность Symbian вызывает только положительные отклики. Даже на смартфонах с ограниченными ресурсами система работает без сбоев и зависаний. Очень хорошо реализована многозадачность, то есть

одновременное выполнение нескольких приложений. Новый Symbian (Symbian 3) будет готов уже в ближайшие месяцы, а вот когда появятся первые смартфоны с ним – неизвестно.

Достоинства:

– Низкие требования к памяти и процессору;

– Функция освобождения неиспользуемой памяти;

– Стабильность;

– Малое количество вирусов для этой платформы;

– Быстро выходят новые версии и исправляются нестабильности;

– Большое количество программ.

Недостатки:

– Для связи с ПК нужно устанавливать дополнительный софт

– Несовместимость программ для старых и новых версий

# 1.3 Windows Mobile


Windows Mobile – общее название нескольких вариантов операционных систем для мобильных устройств, на сегодня является сильнейшим решением в своем роде, для которого выпущено немалое количество программного обеспечения (на фото – Samsung i8000 Witu AMOLED).


Рисунок 1.3 Дизайн Windows Mobile

Главные преимущества Windows Mobile – привычный по настольным ПК интерфейс, хорошая реализация многозадачности, поддержка аппаратов с высоким разрешением экрана, большое разнообразие моделей смартфонов, обилие программного обеспечения на любой вкус и под любые задачи. Сегодняшнее заполнение рынка Windows Mobile – коммуникаторов позволяет пользователю практически в любом ценовом диапазоне найти достойные модели.

Среди программ для устройств в Windows Mobile есть и софт, хорошо знакомый по настольным ПК. С 5-й версии операционной системы появилась новая версия набора офисных программ «Office Mobile»: это знакомые «Word», «Excel», «Outlook», и «Internet Explorer», и «Windows Media

Player».

На данный момент в устройствах используется версия Windows Mobile 6.5, но уже имеется информация о Windows Mobile 7.0 и 7.1. Одной из основных характеристик новой системы станет поддержка управления в режиме multitouch и интегрированное веб-приложение Silverlight для обработки мультимедиа. Доработан также режим блокировки устройства: теперь на экран телефона выводится намного больше информации: о новых SMS, вызовах, времени и дате. Также были оптимизированы выпадающие меню, установлена более легкая в управлении обновленная версия Internet Explorer Mobile.

Стоит отметить, что компания Microsoft представила бету Microsoft Office Mobile. Пакет включает в себя Word Mobile, Excel Mobile и PowerPoint Mobile. В нем реализована поддержка онлайн-приложений, кроме того, улучшены характеристики отдельных мобильных приложений: в Outlook Mobile реализована возможность отображения переписки в виде разговора и группировка связанных между собой писем, в, Excel Mobile есть возможность масштабировать таблицы и разворачивать их на весь экран, также с помощью Mobile Document Viewer можно проще работать с файлами, расположенными на удаленных серверах.

Достоинства:

– Схожесть с настольной версией

– Удобная синхронизация

– В комплекте идут офисные программы
– Многозадачность

Недостатки:

– Высокие требования к оборудованию
– Наличие большого числа вирусов
– Нестабильности в работе

## 1.4 Android

Android – одна из самых молодых мобильных ОС, основанная на операционной системе Linux, платформа для мобильных телефонов, разрабатываемая Open Handset Alliance (OHA). Разработка инициирована компанией Google (на фото – Samsung i5700 Galaxy Spica).


Рисунок 1.4 Дизайн Android

Основным преимуществом Android по сравнению с другими операционными системами является практически полностью открытая архитектура и глубокая интеграция с сервисами Google. Уже сейчас существует достаточно большое количество программного обеспечения для этой ОС, чтобы практически любой пользователь не почувствовал себя обделенным, при этом рынок ПО развивается достаточно динамично.

Работать с этой операционкой удобно, она отлично подходит именно для таких устройств как коммуникаторы, самый большой плюс которых – использование онлайновых сервисов. Прогноз погоды, котировки акций, новости – со всем этим вы можете ознакомиться посредством вашего мобильного устройства. Плюс ко всему удобно реализована работа

с социальными сетями, файловыми сервисами.

## 1.5 BlackBerry OS

**BlackBerry** – это торговая марка беспроводного ручного устройства, которое было впервые представленное в 1997 году компанией Research In Motion. Основная функция – мгновенное корпоративное общение. Главное отличие смартфона BlackBerry – это моментальная синхронизация с корпоративным почтовым сервером. Пользователь получает почту на смартфон непосредственно в момент ее поступления на корпоративный почтовый ящик. При этом обеспечена надежная защита данных с помощью уникальной системы шифрования. Объем передаваемого траффика – минимален, что актуально в роуминге. Серверы, через которые предоставляется сервис защищенной почты находятся в Америке и Англии. Этими смартфонами пользуются в ос-

новном компании и не мелкие, т.к. удовольствие не дешевое. По поводу исключительной защищенности есть и ложка дегтя – публикации, о том, что многие спецслужбы получили коды шифрования от этой системы.

Основной функционал операционной системы Blackberry OS заточен под офисного, бизнес-пользователя Полная QWERTY-клавиатура, специально спроектированная для набора большими пальцами рук, «прокрутка» содержимого экрана и возможность копировать из других сообщений и из Интернета делают написание писем особенно быстрым и удобным. Стандартные приложения для смартфонов BlackBerry позволяют просматривать вложенные файлы большинства основных форматов и работать с ними.


Рисунок 1.5 Дизайн BlackBerry OS

В «Приложениях» находятся текстовый и графический редакторы, редактор презентаций полностью совместимых с настольными офисными программами. **Это весомое конкурентное преимущество**, т.к. в других ОС офисные программы необходимо приобретать, они занимают оперативную память (не встроены в ОС) и не всегда стабильно работают. Планшет BlackBerry PlayBook станет первой моделью, созданной специально для корпоративных пользователей, сочетая в себе полную многозадачность и высокую производительность при работе с мультимедиа. Операционная система BlackBerry Tablet OS была создана специально для планшетных компьютеров.

Новый браузер на основе Webkit открытого движка (как и конкуренты Safari, Google Chrome и др.) поддерживает увеличение отдельных участков страниц при помощи жестов и одновременную работу нескольких сессий (при помощи закладок), а также обладает высокой эффективностью, то есть для его работы требуется меньший объем загружаемых данных.

При создании операционной системы BlackBerry были сохранены все преимущества платформы BlackBerry и в дополнение к функционалу для работы в Интернет, включили в состав новой операционной системы множество новых мультимедийных приложений, простых и удобных в использовании и интегрированных с другими функциями смарт-

фона. Кроме того, в новой версии операционной системы появились такие функции, как Ленты новостей социальных сетей (Social Feeds) и Универсальный поиск (Universal Search), которые дополнительно расширяют и без того богатый спектр возможностей для общения.

## 1.6 iPhone OS

**Apple iOS** (ранее называвшаяся **iPhone OS**) – операционная система, разработанная компанией Apple на основе стационарной Mac OS X для мобильных устройств: iPhone, iPod Touch, iPad.

Сегодня это лидер рынка во многих странах, но в Азии и Европе, как и в России все еще сильны позиции устаревающей Symbian, а США в спину дышит молодая поросль Android и офисная Blackberry.

Рисунок 1.6 Дизайн iPhone OS

**iOS** – это полностью закрытая платформа, которая заставляет пользователя приобретать ее продукцию как софт так и хард, включая многочисленные аксессуары, а также контент и предоставляющая взамен простоту пользования, дружелюбный интерфейс, работоспособность и прочие добродетели.

**Достоинства:**

– Удобство пользования;

– Качественная служба поддержки;

– Регулярные обновления, устраняющие многие проблемы в работе;

– Возможность купить в App Store множество различных программ

**Недостатки:**

– Необходимость джайлбрейка для установки неофициальных приложений;

– Заблокированный характер ОС;

– Отсутствие многозадачности;

– Нет встроенного редактора документов.

## 1.7 Bada

Bada – собственная система компании Samsung. Она была представлена в феврале 2010 года, а первое устройство на этой ОС – Samsung 8500 Wave было очень успешным рынке. Легко различить у Самсунга устройства на BADA (морская тематика) называются Wave (Волна), на Android (космическая тематика) – Galaxy (Галактика)


Рисунок 1.7 Дизайн Bada

BADA – это скорее мобильная платформа, но при этом не полноценная операционная система.

В смысле развития собственной экосистиемы Samsung идет по стопам Apple, копируя их решения которые даже внешне похожи на яблочные :

– Книжный магазин содержит 60 000 книг и продолжает развиваться, клон Apple Bookstore;

– Сервис Dive позволяет найти телефон с помощью опре-

деления местоположения и закрыть к нему доступ или стереть информацию;

– Socilal Hub позволяет систематизировать работу с социальными сетями, обьединяя контакты, календарь и информацию поступающую от их в единый поток данных, который пользователь получает непрерывно с помощью push-технологий на свое мобильное устройство.

## 1.8 TouchWiz от Samsung


Рисунок 1.8 Дизайн Samsung

Пользовательский интерфейс TouchWiz (модели Samsung SGH-F480 TouchWiz, Samsung s8000 Jet, Samsung WiTu, Samsung M8800 Pixon) появился в результате эволюции ин-

терфейса Croix (на фото – Samsung s8000 Jet).

Последняя версия – 2.0 – более «объемна» по дизайну и унифицирует то, как выглядят на экране различные платформы (Windows Mobile, Symbian, Android), а также организует рабочий стол в так называемый мультимедийный куб (кубический шестисторонний рабочий стол). В последней версии есть три панели для виджетов, которые можно перетягивать по экрану простым перемещением и вытягивать из боковой панели простым движением пальца. Одним движением можно настроить и сам экран (например, выбрать обои, раскрыв Home Screen Customizer), проскроллить основные пункты меню, создать сообщение и т. д.

В TouchWiz 2.0 также поддерживается акселерометр и приложение разблокировки, которое дает быстрый доступ к некоторым апплетам в заблокированном режиме.

## 1.9 Обзор инструментов разработчика приложений для мобильных устройств

Есть два принципиально различных типа программ для мобильных устройств: самостоятельные приложения и исполняемые файлы, которые запускаются только при наличии установленной в устройстве специальной среды – интерпретатора.

В первом случае для «перевода» текста программы на язык, понятный какой-либо платформе (операционной

системе), необходим компилятор – специальное приложение, которое, как правило, входит в состав средств разработчика. Пропускаем написанный код через компилятор и на выходе получаем самостоятельное приложение для совместимой платформы. Достаточно скопировать его на соответствующий аппарат и элементарно запустить. Поясним: в случае с обычной Windows XP компилятор выдает EXE-файл. Все, что требуется от пользователя для запуска, – это двойной клик. Компилируемые языки программирования в освоении сложны, зато творческих возможностей предоставляют больше. C++, например, – стандарт де-факто при разработке ПО, в том числе и для многих мобильных платформ.

Первый метод создания программ отличается инструментами (для каждой операционной системы – свои) и файлы, созданные в этих инструментах, запускаются только на тех платформах, для которых они созданы.

Во втором случае – интерпретатор занимается тем, что объясняет данному устройству, как следует выполнять код программы. Пожалуй, самый известный пример интерпретатора – виртуальная машина Java, которая, кстати, по умолчанию наличествует не только в смартфонах, но и практически в любых современных телефонах. Интерпретатор Java – универсален. Одна и та же Java-программа, как правило, выполняется и на Windows Mobile коммуникаторе, и на каком-нибудь музыкальном телефоне Sony Ericsson.

Ссуществуют интерпретаторы для мобильных приложений, написанных на языках Python, mShell (создан фирмой infowing AG (www.mshell.net)) и Basic, хотя эти интерпретаторы скорее экзотика, чем норма.

Минусы интерпретаторов – в относительно медленной скорости работы, а кроме того, они обладают изрядным аппетитом в плане потребления ресурсов. Зато такие языки просты для изучения и инструменты для их создания носят универсальный характер и созданные программы работают на всех платформах одинаково.

Таким образом, в зависимости от типа (исполнения) программного обеспечения для мобильных устройств можно выделить следующие классы инструментария программиста:

1. Инструменты для разработки «мидлетов» – программ, выполняемых на виртуальных Java машинах мобильных устройств (или программ для других интерпритаторов);

2. Инструменты для создания специализированного программного обеспечения под одну из мобильных операционных систем.

### *1.9.1. Инструменты для разработки «мидлетов».*

На данный момент почти все выпускаемые мобильные устройства имеют предустановленную возможность для запуска Java-программ (мидлетов). Большая распространенность этой технологии привлекает внимание разработчиков коммерческих продуктов (особенно игр), но и обычный пользователь может сделать что-нибудь свое.

Базовый язык для разработки программ под Java интерпретатор («мидлетов») Java ME. Чтобы вести программирование по этой технологии необходимо создать у себя на компьютере специальную среду разработки. Основу этой среды составляет Java ME SDK – специальный комплект средств разработки. В настоящее время существует несколько различных версий **SDK** от разных производителей, их использование позволяет создавать мобильные приложения, заточенные под определенные телефоны и мобильные платформы. Соответственно доступные программисту JSR расширения и функциональные возможности среды разработки будут сильно зависеть от выбранного SDK. Наиболее распространенные Java ME SDK программиста следующие**:**

– Sun Java ME SDK 3.0
– NetBeans 6.5 IDE
– MOTODEV Studio for Java ME
– Nokia S60 SDK
– Nokia S40 SDK
– Nokia NFC SDK
– BlackBerry JDE 4.7
– Sony Ericsson SDK 2.5 for Java ME
– LG SDK 1.2 for Java ME

Кроме того, для разработки «мидлетов» применяются специальные интегрированные среды, например – MIDletPascal.

**Sun Java ME SDK 3.0**

**Sun Java ME SDK 3.0** стала де-факто стандартом на рынке мобильных программ. **Java ME SDK –** кульминация проекта Java Wireless Toolkit. **J2ME SDK** поддерживает следующие JVM платформы:

– CLDC/MIDP: Общая JVM конфигурация для мобильных телефонов.

– CDC/FP/PBP/AGUI: JVM конфигурация для high-end смартфонов

– CDC/FP/PBP/BD-J: JVM конфигурация для Blu-ray Disc плееров.

**Java ME SDK –** одна из нескольких доступных SDK ориентированных на некое гипотетические устройство, что дает возможность разрабатывать и отлаживать мобильные приложения перед «заточкой» их под конкретную мобильную платформу. SDK содержит Platform Manager, который позволяет эмулировать конкретную платформу. На рисунке 1.9 показан **Java ME SDK 3.0** с запущенным эмулятором JavaFX телефона.

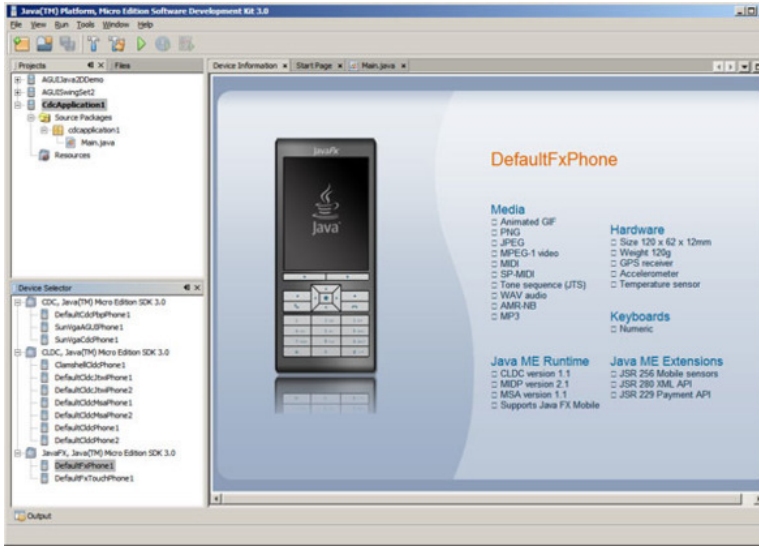



Рисунок 1.9 Вид среды разработки Sun Java ME SDK 3.0

В отличие от Java Wireless Toolkit, **Java ME SDK** содержит IDE и Вы можете разрабатывать и тестировать свои приложения в этой среде. Нужно отметить, что **Java ME SDK** не поддерживает разработку JavaFX приложений, однако он содержит несколько эмуляторов JavaFX 1.1 телефонов (один с тачскрином и один – без), которые позволяет запускать и тестировать JavaFX Mobile приложения. Для создания JavaFX Mobile приложений можно использовать NetBeans IDE. Основным отличием Java ME SDK 3.0 от предыдущих версий является процесс конфигурирования SDK для Blu-

ray разработки. Последняя сборка содержит BD-J библиотеки. Таким образом, устранены преграды, стоявшие перед разработчиками BD-J приложений.

Одной из главных особенностей **Java ME SDK 3.0** является возможность пошагово отлаживать приложения на реальном мобильном устройстве. Данная возможность пока доступна только для Windows Mobile 6 устройств.

**MOTODEV Studio for Java ME**

**MOTODEV Studio for Java ME** – еще одна Java ME SDK, ориентированная на **Motorola** устройства и имеющая ряд дополнительных сервисов:

– Bluetooth Service
– Landmark Storage
– Location Service
– Remote Control (Bluetooth)
– SIM Configuration
– SIP Proxy
– WMA Server

Эти сервисы позволяют Вам симулировать реальные события без необходимости отладки на реальном устройстве. Например, Bluetooth Service содержит Rococo Bluetooth симулятор, который позволяет симулировать Bluetooth устройства в **MOTODEV Studio**.

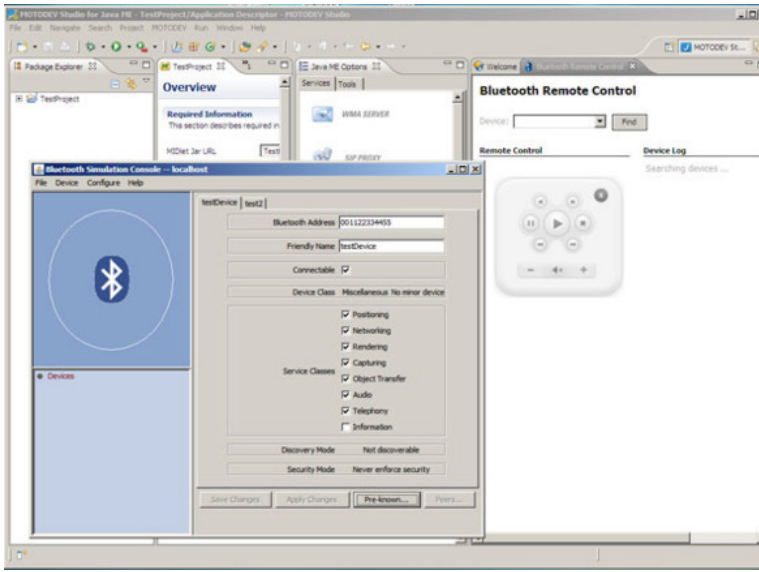



Рисунок 1.10 Вид среды разработки MOTODEV Studio for Java ME

**MOTODEV Studio** отлично подходит для разработки приложений ориентированных на Motorola утстройста. Вы можете отлаживать приложения на реальных устройствах подключив их по USB.

**Nokia S60, S40, and NFC SDK**

**Nokia** предлагает программистам 3 SDK для разработки мобильных приложений. В состав **SDK** различные утилиты, например SVG => SVG-Tiny конвертор, который мо-

жет быть очень полезным, если Вы планируете использовать JSR 226 API для отображения векторной графики. Как и рассмотренные выше SDK, **S60 SDK** позволяет проводить отладку приложений на реальных устройствах, однако он имеет особенность, позволяя перенаправлять System. out и System.err сообщения.

**S40 SDK** включает Nokia Connectivity Framework, который позволяет эмулировать Bluetoth и SMS сообщения.

Если Вы хотите заняться разработкой для wireless smart card, Вам стоит задуматься над использованием инструментов **S40 Nokia 6212 NFC SDK.**

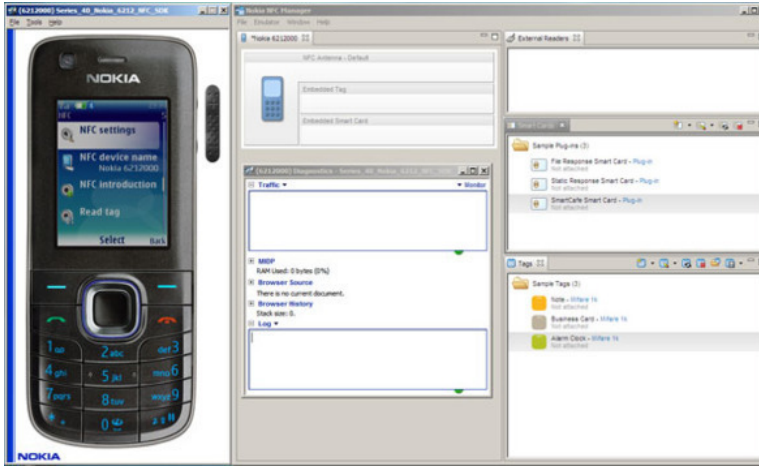



Рисунок 1.11 Вид среды разработки Nokia

Этот SDK не только поддерживает JSR 257 API, но и позволяет симулировать наличие либо отсутствие виртуальной смарт карты. SDK также поддерживает OMNIKEY и PEGODA карт-ридеры, которые подключены к Вашему настольному компьютеры, что позволяет быстро создавать и тестировать приложения на реальных NFC картах. Скриншот S40 Nokia 6212 NFC SDK показан выше.

**BlackBerry JDE 4.7**

**BlackBerry JDE 4.7** – это полноценная среда для разработки и тестирования мобильного приложений для **BlackBerry.** Чтобы помочь разработчикам с их проектами, **BlackBerry JDE 4.7** содержащих более 50 примеров проектов, которые используют **Java ME JSR API** и дополнительные **BlackBerry API.** JDE 4.7 содержит эмуляторы BlackBerry 9500/9530 с сенсорным экраном.

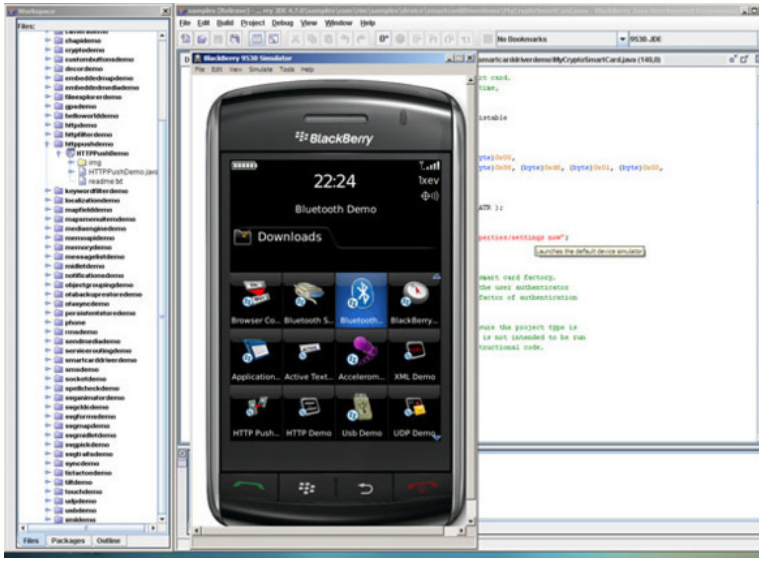



Рисунок 1.12 Вид среды разработки BlackBerry JDE 4.7

Кроме того эмулятор может реагировать на следующие события:

– Наличие USB соединения
– Наличие гарнитуры
– Эмуляция сенсорного-скрина
– Изменение ориентации (тряска устройства)
– Уровень батареи
– Установка или извлечение SD карты
– Входящий звонок
– Изменение GPS положения

– Использование камеры

**Sony Ericsson SDK 2.5 for Java ME**

Если Вы хотите сосредоточить свое внимание над экспериментами с JSR расширениями, можете поиграться с **Sony Ericsson SDK 2.5 for Java ME**. Особенно Вам следует обратить внимание на этот SDK, если Вы хотите использовать JSR 177 Security или Trust Services API (SATSA):

– SATSA APDU: Базовые соединения с Java Card апплетами на SIM карте

– SATSA Crypto: Для шифрования

– SATSA PKI: Цифровая подпись

– SATSA JCRMI: Для RMI соединения с Java Card апплетами на SIM картами

**Sony Ericsson SDK 2.5 for Java ME** поддерживает 3D графику и анимацию: JSR 184 (Mobile 3D Graphics), JSR 239 (Java Binding for OpenGL ES) и Mascot Capsule API. Sony Ericsson SDK 2.5 for Java ME один из нескольких SDK, которые поддерживают JSR 229 Java Payment API. На приведенном ниже рисунке показана интеграция **Sony Ericsson SDK 2.5** в NetBeans 6.5 IDE.

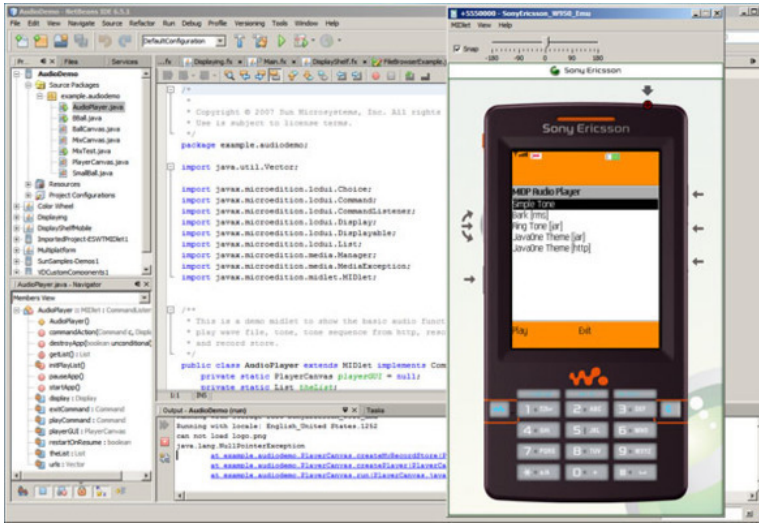



Рисунок 1.13 Вид среды разработки Sony Ericsson SDK 2.5 for Java ME

**LG SDK 1.2 for Java ME**

**LG SDK 1.2 for Java ME** не блещет особой функциональностью и не очень хорошо поддерживает JSR расширения.

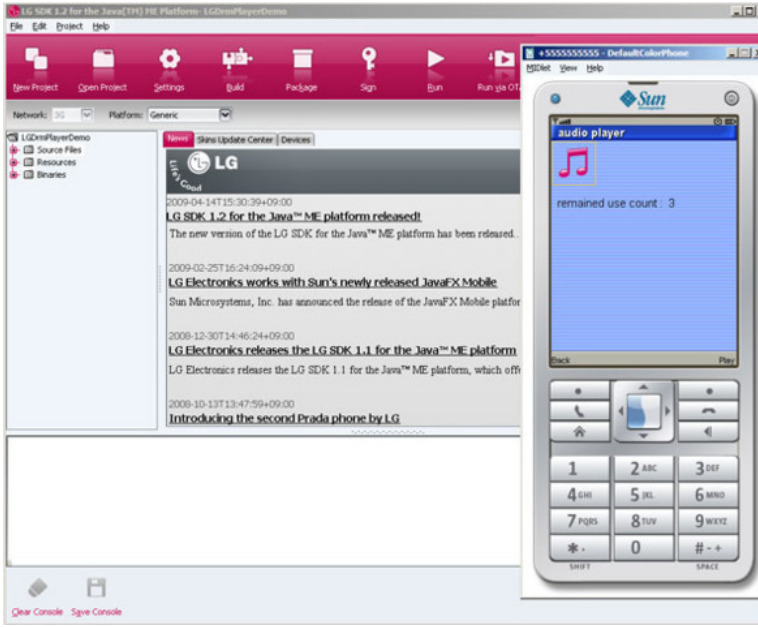



Рисунок 1.14 Вид среды разработки LG SDK 1.2 for Java ME

Однако, это единственный SDK с поддержкой JSR 300, и DRM API, которые обеспечивают работу с защищенным цифровым контентом (графикой, звуком, видео). **LG SDK 1.2 for Java ME** не содержит IDE, однако он, как впрочем и все другие SDK, может использоваться с NetBeans IDE.

**LG SDK 1.2** может симулировать различные события:

– Изменения в файловой системе

– Изменение местоположения
– Транзакция оплаты
– Изменение состояние подключенного устройства

**LG SDK 1.2** содержит также средства просмотра SVG файлов.

Для более удобного программирования в указанных выше SDK, удобно применять интегрированные среды разработчика (IDE), имеющие инструменты визуального программирования форм и встроенные отладчики. Наиболее применяемой IDE для создания «мидлетов» является среда **NetBeans.**

**NetBeans 6.5 IDE**

Если вы хотите поработать над визуальным аспектом своего приложения, то вам следует воспользоваться **NetBeans IDE**. Эта среда наиболее подходит для разработки, проектирования и тестирования JavaFX приложений. Основным принцип JavaFX – дать разработчикам возможность разрабатывать десктопные, веб-ориентированные и мобильные приложения, используя один API framework.

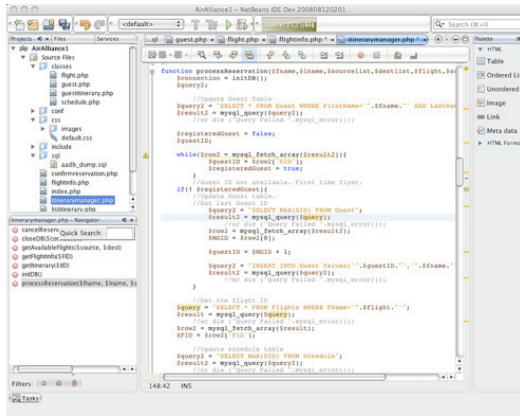


Рисунок 1.15 Вид среды разработки NetBeans 6.5 IDE

В состав **NetBeans 6.5 IDE** входит Mobility Visual Designer – WYSIWYG утилита, позволяющая в визуальном режиме проектировать интерфейс Вашего приложения. В состав NetBeans 6.5 IDE (org.netbeans. microedition) входят следующие визуальные компоненты: Alert, File Browser, Form, List, Login Screen, PIM Browser, SMS Composer, Splash Screen, Text Box, Wait Screen.

Mobility Visual Designer поддерживает векторную SVG графику и анимацию.

Язык Java показался сложным? Тогда стоит использовать программы-посредники: «скармливайте» им программу, написанную на родственниках таких популярных язы-

ков, как Pascal и Basic, и на выходе получайте готовый Java-мидлет. Ярким примером такого «посредника является IDE **MIDletPascal.**

**MIDletPascal**

MidletPascal – это инструмент (IDE) для написания программ на языке Pascal для мобильных телефонов. Код транслируется в привычные для владельцев мобильников JAD и JAR файлы. Поставляется MidletPascal с собственной, дружественной к пользователю средой разработки (IDE). Среда имеет встроенный компилятор, инспектор кода Java и обеспечивает построение архива JAR, что избавляет от установки Java SDK. В итоге компиляция и компоновка мидлетов проста, как нажатие на кнопку. Порадует вас встроенная справка по доступным функциям: работа с графикой, SMS, звуками, файлами и т. д.

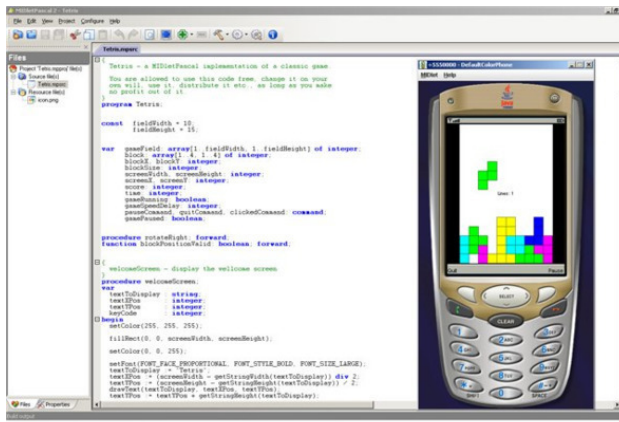


Рисунок 1.16 Вид среды разработки MIDletPascal

Таким образом, для компиляции не требуется ничего, кроме самого MidletPascal, что очень удобно при начальном обучении программированию для сотовых телефонов.

**OmegaBasic**

OmegaBasic – специализированная среда разработки для создания программ и игр. Включает удобный редактор, поддерживает работу с проектами. Можно просматривать несколько файлов, ставить закладки, распечатывать справку по функциям. OmegaBasic позволяет работать с графикой, картами, звуком, музыкой, анимацией, видео и сетью. В качестве основного языка программирования исполь-

зуется Basic. Можно программировать и на Java, используя расширенный набор API OmegaBasic. Сайт разработчика – omegabasic.thegamecreators.com, оттуда можно скачать триальную версию, в которой программы ограничиваются 250 строками кода. Для функционирования OmegaBasic необходимо установить на ПК Java 1.4 SDK и Java Mobile 1.2 SDK.

**MobileBasic**

MobileBasic – такая же специализированная среда. Ее особенность – наличие сервисов по так называемой немедленной OVER-THE-AIR («по воздуху») установке написанных мидлетов на телефоны. Написав программу, программист в MobileBasic может сохранить ее в виде JAD и JAR – файлов. Эти файлы с помощью MidletUploader выгружаются на сервер MobileBasic. Сервер создает WAP/WML – страницы, подключившись к которым с помощью WAP-браузера телефона можно установить мидлет. Кроме этого сервиса, в MobileBasic имеется графический редактор, редактор карт и плиточных изображений, а также редактор мелодий для телефонов Nokia.

Скачать демо-версию MobileBasic можно со страницы www.mobilebasic.com/desktopedition.html. Стоимость продукта – 24,99 фунтов стерлингов, ограничение триальной версии – максимум 1 Кб исходного кода. На сайте не стоит пренебрегать регистрацией, иначе запустить MobileBasic получится не более 30 раз. Как и в прошлом случае, необходи-

мы установленные на компьютер Java SDK.

**Итог:** OmegaBasic и MobileBasic – почти близнецы в плане подхода к написанию кода, к тому же обладают схожим функционалом. Недостатки: необходимость приобретать платную версию и устанавливать Java SDK. На этом фоне ярко выделяется MidletPascal – самый популярный, простой в установке, и главное – бесплатный! Именно поэтому среда MidletPascal, в данном пособии, будет в дальнейшем рассматриваться более подробно.

### *1.9.2 Инструменты для создания программного обеспечения в операционных системах мобильных устройств*

**Инстументарий для программирования в Symbian**

Программная платформа Symbian Series 60 (или S60) – самая популярная в мире смартфонов и коммуникаторов, если судить по продажам мобильных устройств. Поэтому приложения именно для этой платформы весьма актуальны.

C++ for Symbian – наилучший (и, по сути, единственный) язык для создания профессиональных и коммерческих приложений для смартфонов Symbian Series 60. Именно на нем пишется сама система и предустановленное программное обеспечение. Если вы полны решимости программировать на C++ for Symbian, то необходимо установить:

– среду разработки – CodeWarrior (www.forum.nokia.com/codewarrior),

– Carbide. c++ (www.forum.nokia.com/main/resources/tools_and_sdks/ carbide_cpp/) или другое;

– SDK для Symbian соответствующей Edition и Feature Pack под нужную среду разработки (www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html);

– Java 2 Standard Edition;

– Perl версии не ниже 5.003.07.

При создании программ на C++ for Symbian можно получить доступ ко всем возможностям смартфона, что не идет ни в какое сравнение с Java и прочими интерпретируемыми языками. Полученные продукты будут потреблять минимум ресурсов и работать с максимальной скоростью, так как между программой и системой не будет посредников-интерпретаторов.

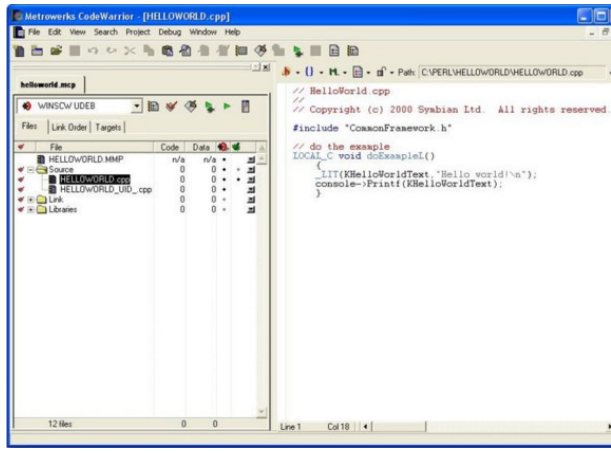



Рисунок 1.17 Инстументарий для программирования в Symbian

При компиляции программного кода создается не только приложение, а также иконка и необходимые файлы ресурсов. Все это в конце концов упаковывается в установочный файл SIS. После его подписи (если необходимо) разработчик может распространять и продавать программу как готовый продукт.

**Инстументарий для программирования в Windows Mobile**

Для создания программ под Windows Mobile Microsoft предлагает среду разработки Visual Studio. Если у вас уже есть установленная Visual Studio 2010 (Professional или

Ultimate), то вы можете использовать для разработки свою редакцию Visual Studio 2010 после установки Windows Phone Developer Tools.

Также существует Expression Blend for Windows Phone – программа для разработки дизайна, которая позволяет создавать и добавлять специальные визуальные возможности, такие как градиенты, анимации и переходы. Для некоторых задач Expression Blend проще в использовании, чем Visual Studio. Следующий список содержит некоторые задачи, которые легко выполняются с помощью Expression Blend.

– Визуальное создание шаблонов данных

– Использование во время разработки тестовых данных для визуализации шаблонов данных

– Визуальное создание стилей элементов управления

– Создание и просмотр анимации

На следующем изображении показан внешний вид Expression Blend.

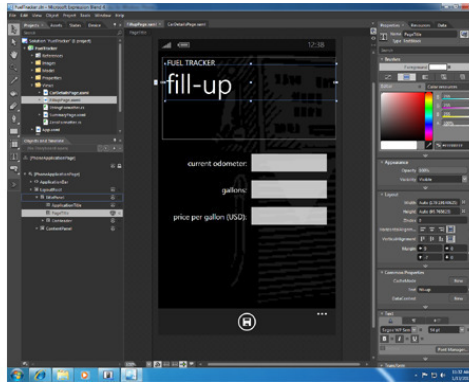



Рисунок 1.18 Среда программирования Expression Blend

Имеется бесплатный пакет: Visual Studio 2010 Express for Windows Phone, который включает в себя drag-and-drop дизайнер, эмулятортелефона, редактор кода и отладчик. Если вы работали с Visual Studio для разработки других видов приложений, вы обнаружите среду для разработки мобильных приложений очень знакомой. На следующем изображении показан внешний вид Visual Studio 2010 Express for Windows Phone.

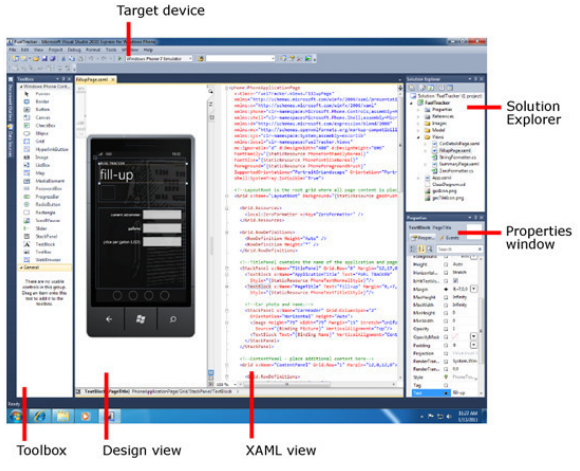



Рисунок 1.19 Среда программирования Visual Studio Phone

Дизайнер для Windows Phone содержит панель инструметов (Toolbox), режим дизайна (Design view), режим XAML (XAML view), обозреватель решений (Solution Explorer) и окно «Свойства» (Properties window), похожие на стандартный дизайнер Visual Studio. Два ключевых различий в том, что в режиме дизайна поверхность выглядит как Windows Phone устройство, и появилось целевое устройство (Target device), которое позволит вам выбрать, будет ли вы отлаживать приложение на устройстве или эмуляторе.

**Инстументарий для программирования в Android**

Самый простой способ приступить к разработке приложений для Android – это загрузить SDK Android и Eclipse IDE. Разработку Android-приложений можно вести на платформах Microsoft Windows, Mac OS X или Linux. Чаще всего используется Eclipse IDE и плагин Android Developer Tools для Eclipse.

Android-приложения пишутся на языке Java, но компилируются и выполняются в Dalvik VM (не в виртуальной машине Java). Кодирование на языке Java в рамках Eclipse – интуитивно понятный процесс. Eclipse предоставляет богатую среду Java, включая контекстно-зависимую справку и подсказки к коду. Когда ваш Java-код будет безошибочно скомпилирован, Android Developer Tools сам позаботится о том, чтобы приложение был надлежащим образом упаковано, в том числе снабдит его файлом AndroidManifest. xml.

Android-приложение можно написать и без Eclipse и плагина Android Developer Tools, но для этого нужно хорошо разбираться в Android SDK.

Android SDK распространяется в виде файла ZIP, который распаковывается в папку на жестком диске. Так как вышло несколько обновлений SDK, рекомендуется поддерживать среду разработки в порядке, чтобы можно было легко переключаться между разными установками SDK.

Android-приложения могут тестироваться как на реальном устройстве, так и на эмуляторе Android, который прилагается к SDK Android. На рисунке показан главный экран

эмулятора Android.

Отладочный мост Android – утилита adb поддерживает несколько дополнительных аргументов командной строки, которые обеспечивают мощные функции, такие как копирование файлов в устройство и из него. Аргумент оболочки командной строки позволяет подключаться к самому телефону и подавать простые команды оболочки. Рисунок 4 иллюстрирует команду оболочки adb, подаваемую реальному устройству, подключенному к ноутбуку под Windows с помощью кабеля USB.

**Инстументарий для программирования в BlackBerry**

Разработчики могут использовать как стандартные инструменты на основе веб-технологий, такие как HTML/HTML5, CSS, JavaScript или Java®, так и специальные средства разработки приложений для BlackBerry. Разработка приложений для BlackBerry может вестись также при помощи таких распространенных инструментов, как Eclipse и Microsoft Visual Studio. Таким образом, разработчики обладают свободой выбора наиболее подходящего им инструментария для создания приложений.

**Инстументарий для программирования в Bada**

Предложенный разработчикам инструментарий позволяет писать код не для конкретной ОС, а работать с определен-

ными функциями (например, оболочкой, контактами) а ОС с которыми работает эта надстройка может быть несколько, В рамках бета-версии SDK доступ осуществлялся только к функциям TouchWiz 3D и ряду системных функций, что позволяет сказать, что фактически Bada – это надстройка к интерфейсу. В будущем развитие средств разработки программ позволит писать полнофункциональные программы, задействующие не только интерфейс, но и другие возможности телефона (не затрагивая ОС, лежащую в основе).

Вспомним, что компания Samsung стала первым производителем, кто транслировал интерфейс TouchWiz с собственных устройств на другие ОС, т.е. сделали этот интерфейс кросс-платформенным, чтобы приучить потребителей к нему. Не важно, какая ОС, важно, что везде потребитель видит один и тот же интерфейс и ассоциирует его с компанией Samsung. BADA позволит быстро реагировать на расстановку сил на рынке операционных систем, не привязываясь ни к одной из них, что является достаточно гибкой стратегией.

Компания Samsung активно взялась за работу с разработчикамипрограммного обеспечения, так что уже к моменту выхода ОС Bada на рынок для нее было доступно большое количество разнообразных программ, игр и виджетов, сконцентрированных на специализированном ресурсе Samsung Apps.

**Инстументарий для программирования iPhone**

До официальной публикации SDK у разработчиков не было возможности легальной разработки native – приложений для iPhone и iPod Touch. Учитывая огромный интерес к iPhone, Apple пошла на компромисс: позволила сторонним разработчикам создавать так называемые виджеты – приложения, выполняемые в веб—браузере Safari, интегрированном в iPhone и iPod Touch. Основным отличием виджетов от native—приложений является необходимость написания кода не на Objective C, а с использованием стандартных веб—технологий вроде HTML, CSS, JavaScript и AJAX. С точки зрения пользователя такое приложение отличается тем, что выполняется в веб-браузере и открывается не путем выбора иконки из главного меню устройства, а при выборе закладки. Для ознакомления с процессом создания и развертывания виджетов для iPhone рекомендую почитать книгу «Professional iPhone and iPod Touch Programming», а также заглянуть на http://developer.apple.com/webapps/.

Отсутствие официальной возможности создавать ПО не остановило энтузиастов. Они подготовили средства разработки, позволяющие создавать полноценный софт для JailBreaked iPhone. В процессе JailBreaking на аппарат устанавливается софтина с немудреным названием Installer. С ее помощью пользователи могут скачивать и устанавливать необходимый софт из каталога, который формируется из репозиториев (их адреса прописываются вручную в Installer).

Так что JailBreaking – не только разлочка, но и процедура, позволяющая получить полный доступ к файловой системе iPhone. Описание процесса без труда можно найти в Сети, поэтому мы не будем на этом останавливаться. Софт, распространяющийся через Installer, написан с использованием «неофициального» процесса разработки. До недавнего времени иного пути создания и даже установки стороннего ПО в iPhone не было.

Но в марте 2010 года Apple осчастливили-таки общественность публикацией первой беты SDK. С тех пор на офсайте разработчиков Apple периодически публикуются новые версии беты SDK и документации (на момент написания пособия наиболее актуальной была восьмая). SDK представляет собой IDE XCode, набор необходимых библиотек, эмулятор и другие инструменты.

Чтобы программировать под iPhone, нужен Mac с установленной Mac OS X Leopard. Грустно, но это так. Вообще говоря, можно развернуть среду разработки на Unix и даже пытаться писать из-под VMWare, но это связано с рядом сложностей, которые бурно обсуждаются в интернете. Кроме того, необходимо установить и сконфигурировать SDK. Описание процесса настройки рабочей станции для «неофициальной» разработки можно прочитать в замечательной книжке «iPhone Open Application Development», которую легко найти в Сети.

При разработке приложений для iPhone OS, а также

MacOS 10.5 и выше используется язык программирования Objective C 2.0. Он является своеобразной надстройкой над ANSI C, предназначенной для гибкого объектно-ориентированного программирования. Не совсем понятно, чем Apple не угодил C++. Многие концепции Objective C заимствованы у одного из первых объектно-ориентированных языков Smalltalk. Тем не менее, программа для iPhone может содержать как код на Objective C, так и на C или C++.

При компиляции используются инструменты GNU Compilers Collection, которые распознают принадлежность кода к конкретному подвиду GNU C/C++ по расширению файла. В частности, C – код содержится в файлах с расширением *.c; C++ – код в *.mm; Objective C – в *.m.

Как видно из вышесказанного, одно перечисление (и то – не всех) технологий и инструментов разработки программного обеспечения для мобильных устройств занимает достаточно много места, а для подробного освещения этих технологий требуется не один том учебной литературы. В нашем курсе мы познакомимтя с наиболее общей технологией создания приложений мобильного мира – технологией создания мидлетов. С остальными технологиями (по необходимости) программист, имеющий навык создания мидлетов, сможет освоится сам.

Успехов Вам в изучении курса!

# 2 Разработка приложений для Windows Phone

Windows Phone 7 обеспечивает поддержку двух популярных (в настоящее время), платформ разработки, Silverlight и XNA. Это гарантирует, что в Windows Phone 7 найдется много интересного и для разработчиков.

Silverlight – браузерное развитие Windows Presentation Foundation (WPF) – уже обеспечил Веб-разработчиков беспрецедентными возможностями разработки сложных пользовательских интерфейсов, предоставляя традиционные элементы управления, высококачественный текст, векторную графику, мультимедиа, анимацию и привязку данных, которые могут выполняться во множестве браузеров и на разных платформах. Windows Phone 7 расширяет использование Silverlight на мобильные устройства.

XNA (три буквы, обозначающие XNA «не аббревиатура») – это игровая платформа Майкрософт, поддерживающая основанную на спрайтах 2D графику и 3D графику с традиционной архитектурой игрового цикла. Несмотря на то, что главным предназначением XNA является написание игр для консоли Xbox 360, разработчики могут создавать на XNA программы и для ПК, и для стильного аудиоплеера Майкрософт Zune HD.

Для разработки приложений под Windows Phone жела-

тельно иметь Visual Studio 2010 с Service Pack 1 редакции Professional или выше и пакет разработчика Windows Phone SDK 7.1 (это инструментарий на начало 2012 года, в дальнейшем – номера версий будут меняться). Взять Service Pack 1 для VS-2010 (бесплатно) можно по адресу: http://go.microsoft.com/fwlink/?LinkId= 210710, а Windows Phone SDK 7.1 (тоже бесплатно) – с адреса http://go.microsoft.com/fwlink/?LinkID=226694.

Если у вас нет Visual Studio 2010 Professional, то пакет Windows Phone SDK 7.1 вы все-равно можете установить себе на ПК. В момент установки вам автоматически будет установлена бесплатная версия Visual Studio 2010 Express for Windows Phone, на которой также можно разрабатывать приложения под Windows Phone (при этом, конечно, ваш ПК должен быть связан с интернетом).

Обе версии интегрированных средств разработки Visual Studio предоставляют разработчику полноценные возможности по отладке на устройстве и эмуляторе такие же, какие есть у разработчиков приложений под настольную версию Windows.

Обратите внимание, что для того, чтобы отлаживаться на реальном устройстве, помимо собственно устройства и кабеля для подключения его к компьютеру разработчика, на компьютере со средствами разработки необходимо иметь установленное ПО Zune (http://zune.net). Также перед развертыванием приложения и отладкой, требуется зарегистри-

ровать устройство или «разлочить», с использованием утилиты Windows Phone Developer Registration Tool, которая устанавливается вместе с Windows Phone SDK.

## 2.1 Windows Phone SDK

Этот пакет содержит всё необходимое, для того, чтобы начать разработку. На момент написания этого методического пособия, последняя версия инструментария доступна в версии Windows Phone SDK 7.1 Release Candidate в лицензии «Go Live» с возможностью разрабатывать свои приложения и публиковать их в Windows Phone Marketplace.

Windows Phone SDK 7.1 Release Candidate содержит следующие компоненты:

– Windows Phone SDK 7.1
– Windows Phone Emulator
– Windows Phone SDK 7.1 Assemblies
– Silverlight 4 SDK and DRT
– Windows Phone SDK 7.1 Extensions for XNA Game Studio 4.0
– Expression Blend SDK for Windows Phone 7
– Expression Blend SDK for Windows Phone OS 7.1
– WCF Data Services Client for Windows Phone
– Microsoft Advertising SDK for Windows Phone

Если у вас не установлена версия Visual Studio 2010 редакции Professional, Expression Bland 4 или XNA Game Studio

4.0, в процессе установки также будут скачаны и установлены:

– Visual Studio 2010 Express for Windows Phone
– Expression Blend 4 for Windows Phone
– XNA Game Studio 4.0

## 2.2 Expression Blend и Expression Blend for Windows Phone

Expression Blend – это интерактивный визуальный дизайнер для XAML, технологии описания интерфейса для приложений Silverlight и Windows Presentation Foundation (WPF). Это отличное средство разработки, которое позволяет просто манипулировать слоями, анимацией, стилями и шаблонами. Это базовое средство разработки на XAML. Собственно программа Expression Blend не бесплатна, однако, специальная версия для создания дизайнов приложений под Windows Phone, под названием Expression Blend 4 for Windows Phone доступна для разработчиков бесплатно. Она закачается и установится в процессе установки Windows Phone SDK, если у вас на компьютере нет полной версии Expression Blend. Подробнее об Expression Blend 4 можно прочитать на MSDN: http://msdn.microsoft.com/ru-ru/library/cc296227.aspx

## 2.3 XNA Game Studio 4.0

XNA Game Studio – это программное окружение, которое позволяет разрабатывать в Visual Studio игры для Windows Phone, консоли Xbox 360 и компьютеров на базе Windows. Включает в себя XNA Framework, представляющий собой набор библиотек на управляемом коде для разработки игр. Подробнее можно прочитать на MSDN: http://msdn.microsoft.com/ru-ru/library/bb200104.aspx

## 2.4 Windows Phone Emulator

Несмотря на то, что Windows Phone Emulator не содержит полного набора приложений доступных на реальном устройстве, он предоставляет мощную среду, позволяющую практически полностью разработать приложение в эмуляторе.

Эмулятор Windows Phone Emulator не поддерживает проигрывание медиаконтента Zune. Эмулятор поставляется только с одним встроенным приложением Internet Explorer, но это Internet Explorer 9 с поддержкой HTML5.

При этом эмулятор позволяет тестировать звонки и отсылку SMS сообщений, поддерживает мультитач на мониторах с поддержкой мультитач, поддерживает симуляцию камеры, геолокационных сервисов и акселерометра, а также позволят делать снимки экрана.

Подробнее можно прочитать на MSDN: http://msdn.microsoft.com/ru-ru/library/ff402563(v=VS.92).aspx

## 2.5 Windows Phone Developer Registration Tool

Перед тем, как разработчик сможет развернуть своё приложение на реальном устройстве, его необходимо зарегистрировать как устройство разработчика – «разлочить». Это делается один раз для определенного телефона. Зарегистрированный на Marketplace разработчик может зарегистрировать до 3 устройств (для разработчика, зарегистрированного, как студент количество устройств ограничено до одного). Подробнее: http://create.msdn.com

## 2.6 Windows Phone Profiler

Windows Phone Profiler доступен в меню Debug Visual Studio с установленным инструментарием Windows Phone SDK (рис.).

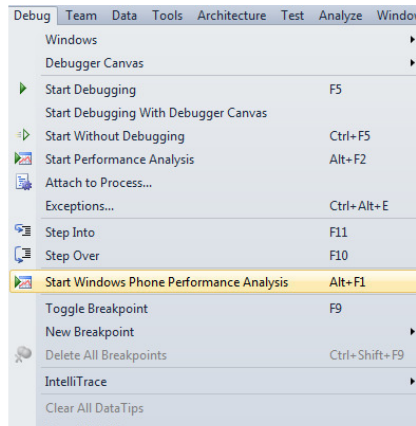



Рисунок 2.1 Меню запуска Windows Phone Analysis

Анализирует работу программы во время исполнения, идентифицирует возможные проблемы с производительностью. Подробнее можно прочитать на MSDN: http://msdn.microsoft.com/ru-ru/library/hh202934(v=VS.92).aspx

## 2.7 Silverlight Toolkit for Windows Phone

Silverlight Toolkit for Windows Phone – набор полезных элементов управления Silverlight дляWindows Phone с поддержкой режима дизайна, от команды разработчиков Silverlight. Доступен весь исходный код, примеры и документация. Обновляется приблизительно раз в три месяца, досту-

пен по адресу http://silverlight.codeplex.com или через NuGet.

Текущий релиз включает в себя такие элементы управления, как ContextMenu, DatePicker и TimePicker, ToggleSwitch, WrapPanel и GestureHelper.

## 2.8 Среда разработки

После установки средств разработки Windows Phone SDK в диалоге New Project в Visual Studio появятся группы проектов для Silverlight for Windows Phone (рис. Рисунок 2.2):

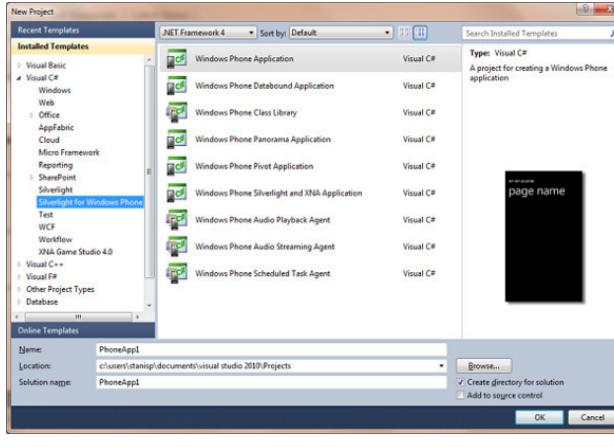



Рисунок 2.2 Группы проектов в меню Silverlight for Windows Phone

и в группе XNA Game Studio 4.0 добавятся проекты для

Windows Phone (рис. 2.2).

Материал данного методического пособия сфокусирован на разработке под Windows Phone на Silverlight, поэтому рассмотрим доступные разработчику приложений шаблоны несколько более подробно.

После установки разработчику доступны следующие шаблоны приложений Silverlight for Windows Phone:

– Windows Phone Application
– Windows Phone Databound Application
– Windows Phone Class Library
– Windows Phone Panorama Application
– Windows Phone Pivot Application
– Windows Phone Silverlight and XNA Application
– Windows Phone Audio Playback Agent
– Windows Phone Audio Streaming Agent
– Windows Phone Scheduled Task Agent

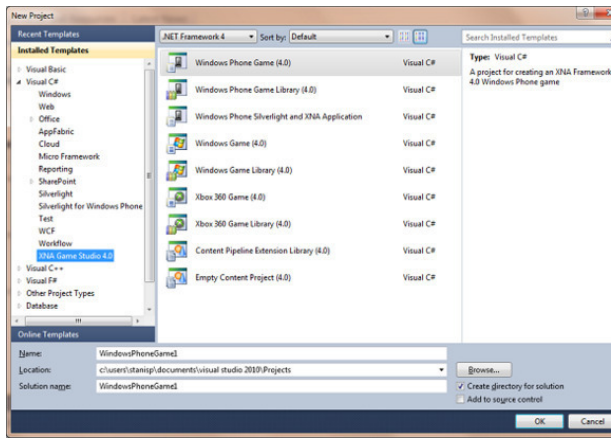



Рисунок 2.3 Шаблоны приложений Silverlight for Windows Phone

Перед тем как перейти непосредственно к шаблонам приложений, надо сказать несколько слов по поводу Windows Phone и Metro-дизайна.

## 2.9 Windows Phone и Metro-дизайн

Платформа Windows Phone не просто очередная платформа для мобильных устройств. Она содержит в себе не только технологическую составляющую, но и полностью проработанную концепцию дизайна интерфейса и взаимодействия с пользователем под названием Metro-дизайн или стиль Metro.

Если вы дизайнер или в вашей команде есть выделенный дизайнер, вы можете воспользоваться всей мощью инструментарий Expression Blend 4 или Expression Blend for Windows Phone, которая поставляется вместе с Windows Phone SDK.

Что же делать если вы разработчик и не хотите заниматься визуальным дизайном приложения, например, вы разрабатываете бизнес-приложение и всё что от него требуется, соответствовать общему дизайну и стилю Windows Phone?

Всё очень просто. Во-первых, Silverlight для телефона разработан с учётом требований Metro-дизайна, поэтому все встроенные элементы управления выполнены в Metro-дизайне. Во-вторых, по умолчанию, приложения, созданные из шаблонов из поставки Windows Phone SDK, работают, выглядят и используют стили и шрифты в соответствии с Metro-дизайном.

С другой стороны, возможностей стилизации элементов управления и приложений, основанных на XAML, которые представляет Silverlight, вполне достаточно, чтобы сделать своё приложение неповторимым и узнаваемым, оставаясь в рамках стиля Metro.

Руководство по дизайну интерфейсов и взаимодействию с пользователем для Windows Phone можно найти по следующей ссылке http://msdn.microsoft.com/ru-ru/library/hh202915.aspx

Всё, что было сказано выше, относится, конечно, к дизай-

ну обычных приложений, так как требования к дизайну игровых приложений и их интерфейсу, могут существенно отличаться. При этом не надо забывать об общих принципах взаимодействия с пользователем, заложенных в концепции Windows Phone.

## 2.10 Шаблоны приложений

Сначала давайте рассмотрим три шаблона, представляющих собой три основных стиля приложения для Windows Phone (рис.):


Рисунок 2.4 Шаблоны приложений

– Windows Phone Application
– Windows Phone Pivot Application
– Windows Phone Panorama Application

Windows Phone Application – это аналог простого диало-

гового приложения, у которого один основной экран, через который происходит основное взаимодействие с пользователем.

Windows Phone Pivot Application – это некий аналог приложения с закладками, где заголовок каждой закладки определяет содержимое. Стандартный вариант использования – каждая закладка представляет собой одни и те же, в целом, данные, но в разных представлениях и/или с разной фильтрацией. Например, календарь, почтовый клиент и настройки телефона. Шаблон использует элемент управления Pivot.

Windows Phone Panorama Application – приложение панорама, в котором зоны взаимодействия с пользователем также разделены на панели, но доступны они через горизонтальную прокрутку; фоновое изображение установлено сразу на всю панораму, она имеет общий заголовок, который прокручивается медленнее, чем панели; контент соседней панели справа виден при отображении текущей. Например, таким образом реализованы хабы в Windows Phone: People, Marketplace, Pictures, Music+Videos и др. Шаблон использует элемент управления Panorama.

Шаблоны, заканчивающиеся на Agent – это шаблоны библиотек, для выполнения соответствующих фоновых задач:

– Windows Phone Audio Playback Agent
– Windows Phone Audio Streaming Agent
– Windows Phone Scheduled Task Agent

Шаблон Windows Phone Databound Application – простой

шаблон приложения с вида список – детальное представление с реализацией навигации между страницами с передачей параметров и хранением данных в глобальном VeiwModel.

Шаблон Windows Phone Class Library – шаблон библиотеки классов для Windows Phone.

Шаблон Windows Phone Silverlight and XNA Application для Silverlight приложения, которое может использовать XNA для рендеринга графического контента.

## 2.11 Создаем первый проект на Silverlight

Прежде чем приступить к приемам программирования для Windows Phone 7, необходимо познакомиться с базовыми понятиями. Те, кто имел опыт программирования на Windows Mobile 6, уже обладают некоторыми знаниями в этой области (использование эмуляторов, отличия от настольной. NET Framework и т.д.). Тем не менее, и им также придется учиться заново, так как Microsoft в очередной раз поменяла правила, и все прежние навыки теперь считаются устаревшими и выброшены на свалку истории. Гонка за новыми технологиями продолжается.

В этом параграфе мы создадим традиционное приложение **Здравствуй, мир!**, чтобы понять основные принципы создания программ для Windows Phone 7. Наше первое приложение будет построено при помощи технологии Silverlight, которая является удобной платформой для бизнес-приложе-

ний и игр.

### *2.11.1 Терминология*

Прежде чем приступить к написанию приложений для Windows Phone, необходимо познакомиться с некоторой терминологией. Рассмотрим некоторые элементы Windows Phone (рис.).

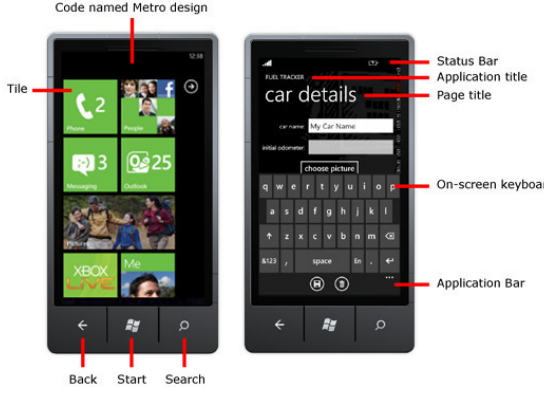



Рисунок 2.5 Элементы интерфейса смартфона

**Tile** (плитка) – Значок приложения на стартовом экране. Плитка может быть динамической и отображать некоторую информацию для пользователя.

**Application Title –** Название приложение. Обычно в верхнем регистре.

**Page Title –** Заголовок страницы. Обычно в нижем регистре.

**Status Bar –** Состояние работы телефонной части, например, уровень сигнала.

**On-screen keyboard –** Экранная клавиатура. Появляется при получение фокуса текстовым полем. Иногда используется термин (SIP – soft input panel).

**Application Bar –** Дополнительная всплывающая панель для навигации по приложению. Содержит кнопки и/или пункты меню.

**Кнопки Back, Start, Search –** Стандартные кнопки на любом устройстве с Windows Phone.

Специально для Windows Phone 7 был разработан новый пользовательский дизайн под кодовым названием Metro. Рекомендуется следовать этому дизайну в своём приложении, чтобы оно интегрировалось с операционной системой и другими приложениями. Дизайн обеспечивает простой в использовании интерфейс, предназначенный для уменьшения потребления энергии на телефоне.

### *2.11.2 Создание нового проекта*

Запустите Visual Studio 2010 Express For Windows Phone. В меню **File** выберите пункт **New Project**. У вас откроется диалоговое окно **New Project**.

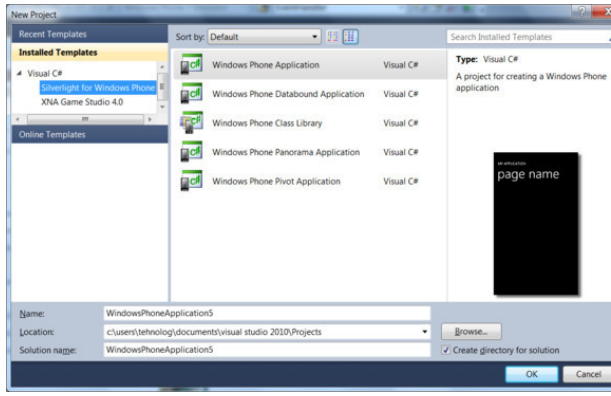



Рисунок 2.6 Окно выбора типа проекта

Далее слева выберите пункт Silverlight for Windows Phone. Как видите, для данного типа проекта доступны несколько шаблонов: Windows Phone Application, Windows Phone Databound Application, Windows Phone Class Library, Windows Phone Panorama Application, Windows Phone Pivot Application. Для нашего учебного примера выберем первый вариант.

Постарайтесь сразу выработать привычку задавать понятные имена для своих проектов. Поэтому присваиваем проекту имя **WP7HelloWorld** и нажимаем на кнопку **OK**.

Спустя несколько секунд Visual Studio создаст новый проект. Вы увидите несколько окон на экране. Оставим пока в покое окна в центральной части экрана с изображением телефона и кодом XAML, а посмотрим на окно **Solution**

**Explorer**. В Solution Explorer хорошо видна структура решения, созданного на основе выбранного шаблона Windows Phone Application. В нашем случае в этом окне содержится один проект **WP7HelloWorld**.

Проект содержит следующие файлы (табл. Таблица 2.1):

*Таблица 2.1*

| | |
|---|---|
| App.xaml/App.xaml.cs | содержит точку входа программы, инициализирует ресурсы програмы и выводит программу на экран |
| MainPage.xaml/MainPage.xaml.cs | содержит страницу с пользовательским интерфейсом |
| ApplicationIcon.png | файл значка в формает PNG, который выводится в списке приложений телефона |
| Background.png | файл изображения в формате PNG, который выводится на стартовой странице |
| SplashScreenImage.jpg | файл изображения, которое выводится во время загрузки приложения. Вы можете заменить на свою картинку |
| Properties/AppManifest.xml | манифест, необходимый для создания сборки |
| Properties/AssemblyInfo.cs | содержит метаданные о имени и версии приложения, которые встраиваются в сборку |
| Properties/WMAppManifest.xml | манифест, который содержит специальные данные о приложении для Windows Phone Silverlight |
| папка References | различные библиотеки (сборки), которые обеспечивают работоспособность приложения |

Подробнее о файлах проекта можно почитать в статье [1].

### *2.11.3 Сборка и тестирование программы*

Хотя наша программа еще бесполезна, тем не менее, давайте проверим ее работу – скомпилируем и протестируем в эмуляторе.

В меню **View** выберите пункт **Output** (возможно потребуется настроить это меню), чтобы открыть окно **Output**. Да-

лее в меню Debug выберите команду **Build Solution** (SHIFT + F6) для компиляции.

Посмотрите в окно Output и изучите сообщения, генерируемые во время компиляции приложения, включая финальное сообщение, в котором подводится окончательный итог и количество предупреждений и ошибок.

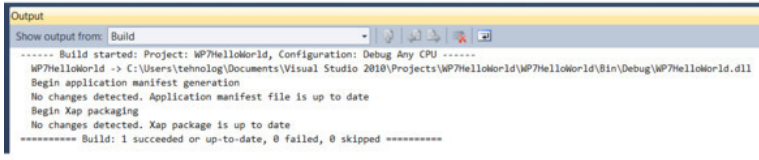



Рисунок 2.7 Вид окна Output

Также вы можете использовать окно **Error List** (View→Other Windows→ Error List), которое показывает ошибки, предупреждения и сообщения, выдаваемые компилятором. Вы можете сделать двойной щелчок на описании ошибки, чтобы автоматически оказаться в нужном месте исходного кода.

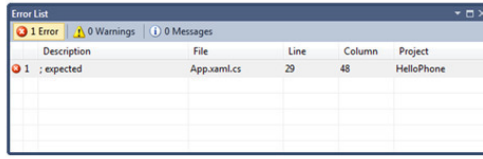



Рисунок 2.8 Вид окна Error List

### *2.11.4 Запуск программы в эмуляторе*

Убедитесь, что у вас установлен **Windows Phone Emulator** в выпадающем списке устройств **Select Device**, который расположен рядом с кнопкой **Start Debugging** на панели инструментов.

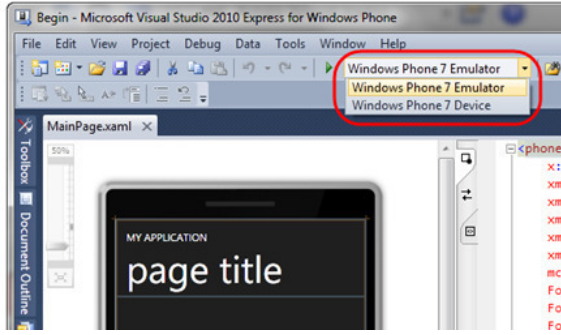



Рисунок 2.9 Список выбора устройств отладки

Нажмите F5 или щелкните по зеленому треугольнику для

запуска программы в Windows Phone Emulator. На экране появится эмулятор устройства и начнется процесс установки приложения на эмулятор. Наберитесь терпения и ждите полной загрузки.

Через некоторое время вы увидите свое приложение в эмуляторе.

Пока программа слишком проста для изучения, поэтому давайте закроем ее – нажмите SHIFT + F5 или щелкните на кнопке Stop на панели инструментов (рис.), чтобы остановить отладчик и закончить сеанс отладки. Но не закрывайте окно эмулятора.

После начала сеанса отладки значительную часть времени занимают настройка среды эмулятора и запуск приложения. Чтобы упростить отладку, не закрывайте эмулятор во время работы с исходным кодом в Visual Studio. Пока эмулятор работает, остановка текущего сеанса, редактирование исходного кода и последующее создание и развертывание нового образа приложения для запуска нового сеанса отладки выполняются очень быстро.

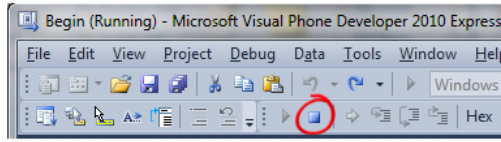



Рисунок 2.10 Кнопка Stop на панели инструментов

### *2.11.5 Проектирование интерфейса пользователя*

Теперь, когда мы поняли, как создавать приложения, попробуем создать более сложную программу с элементами пользовательского интерфейса. Мы добавим такие элементы, как заголовок, текстовое поле и кнопка. При использовании приложения нужно ввести какой-либо текст в текстовое поле, и после нажатия кнопки появится сообщение с введенным текстом. Он будет выглядеть примерно, как на следующем рисунке.

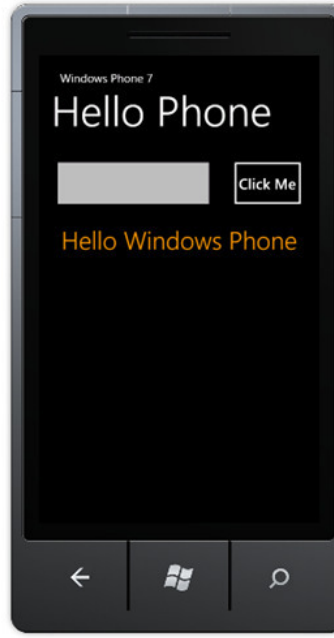



Рисунок 2.11 Результат работы программы в окне эмулятора

В обозревателе решений дважды щелкните файл MainPage. xaml, чтобы открыть его в конструкторе.

Хотя интегрированная среда разработки поддерживает графические манипуляции с объектами (как обычный визуальный конструктор интерфейса), мы вручную отредактируем код XAML. Для перевода режима редактора в представление XAML и увеличения области обзора дважды щелкни-

те вкладку XAML с правого края окна конструктора.

В разметке XAML найдите элемент контейнера **Grid** с именем **LayoutRoot**. Он предназначен для упорядочивания элементов на странице. Внутри его свойства **RowDefinition** вставьте дополнительную строку между двумя существующими и установите значение свойства **Height** равным **Auto**. В этой строке вскоре появится текстовое поле и кнопка.

```
<Grid x: Name=«LayoutRoot» Background=«Transparent»>
<Grid.RowDefinitions>
<RowDefinition Height=«Auto»/>
<RowDefinition Height=«Auto»/>
<RowDefinition Height=«*»/>
</Grid.RowDefinitions>
…
</Grid>
</phone: PhoneApplicationPage>
```

**Grid** – это элемент разметки, который выступает в качестве контейнера для других элементов управления. Его основная задача – расположение и упорядочение дочерних элементов управления. Существует и другие элементы управления разметкой: Canvas, StackPanel.

Обратите внимание, что корневой элемент **Grid** содержит вложенные элементы, каждый из которых сопоставлен отдельной строке внешней сетки путем определения свойства

**Grid.Row**. Найдите элемент **Grid** с именем **TitlePanel**. Присвойте свойству **Text** первого элемента **TextBlock** в пределах внутренней сетки строковое значение **Windows Phone 7**. Аналогичным образом присвойте свойству **Text** второго элемента **TextBlock** строковое значение **Hello Phone**.

Теперь найдите элемент **Grid** с именем **ContentPanel**, который должен быть пустым, и вставьте в него следующую разметку XAML.

```
<Grid x: Name=«LayoutRoot» Background=«Transparent»>
…
<! – ContentPanel – place additional content here – >
<Grid x: Name=«ContentPanel» Grid.Row=«1» Margin=«12,0,12,0»>
<Grid.ColumnDefinitions>
<ColumnDefinition Width=«*» />
<ColumnDefinition Width=«Auto»/>
</Grid.ColumnDefinitions>
<TextBox Grid.Column=«0» Name=«MessageTextBox»
FontSize=» {StaticResource PhoneFontSizeExtraLarge}»
Margin=«20,20,10,20»/>
<Button Grid.Column=«1»
Name=«ClickMeButton»
Content=«Click Me»
HorizontalAlignment=«Right»
Padding=«4»
```

```
Margin=«10,20,20,20» />
</Grid>
</Grid>
…
```

(описание текстового поля можно было также получить, просто перетащив мышкой заготовку текстового поля из панели TOOLS на изображение эмулятора телефона).

Элемент Grid упорядочивает свои дочерние элементы управления на странице на основе ширины каждого столбца, как указано в коллекции ColumnDefinitions. Обратите внимание, что ширина первого столбца указывается как *. Это позволяет столбцу растягиваться и заполнять неиспользуемое пространство в строке после размещения всех остальных столбцов. Ширина второго столбца указывается как Auto, что позволяет изменять размер столбца в соответствии с размером его содержимого.

Чтобы завершить конструирование страницы, добавьте третью строку, которая будет содержать баннер с сообщением, вводимым пользователем. Для создания этой строки вставьте следующую разметку XAML сразу перед конечным тегом внешней сетки.

```
…
<Grid x: Name=«LayoutRoot» Background=«Transparent»>
…
<Grid Grid.Row=«2»>
<TextBlock Name=«BannerTextBlock»
```

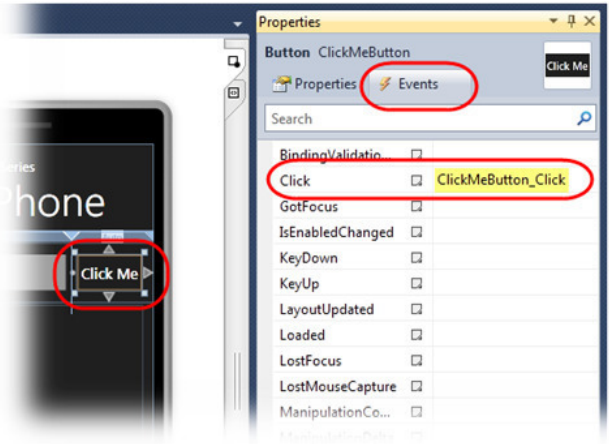
```
Style=» {StaticResource PhoneTextExtraLargeStyle}»
Foreground=«#FFFF9A00»
HorizontalAlignment=«Stretch»
TextWrapping=«Wrap»
TextAlignment=«Center»
FontWeight=«Bold» />
</Grid>
</Grid>
…
```

Щелкните вкладку **Design** с правого края окна, чтобы перейти в режим конструирования.

### *2.11.6 Обработка событий*

Теперь необходимо определить обработчики событий, которые отвечают на действия из интерфейса пользователя, в частности событие нажатия кнопки. Включите в конструкторе режим Design. Для этого дважды щелкните вкладку Design с правого края окна конструктора. Щелкните кнопку **Click Me** на поверхности конструктора, чтобы выбрать ее, а затем нажмите клавишу F4, чтобы открыть окно свойств этой кнопки. На панели **Properties** щелкните вкладку **Events**, чтобы отобразить окно со списком доступных событий. Найдите в этом списке событие **Click** и введите **ClickMeButton_Click** в текстовом поле рядом с этим событием. Нажмите клавишу Enter, чтобы создать обработчик событий с этим именем, и откройте файл с выделенным кодом для отображения заглушки метода, созданной Visual Studio

(рис.).




```
<!--This section is empty. Place new content here Grid.Row="1"-->
<Grid Grid.Row="1">
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="Auto"/>
    </Grid.ColumnDefinitions>
    <TextBox Grid.Column="0" Name="MessageTextBox" FontSize="{StaticResource Pho
    <Button Grid.Column="1" Name="ClickMeButton" Content="Click Me"
            HorizontalAlignment="Right" Padding="4" Margin="10,20,20,20"
            Click="ClickMeButton_Click" />
</Grid>
<Grid Grid.Row="2">
```

Рисунок 2.12 Задание обработчика события Click

Существует альтернативный механизм создания обработ-

чика событий. В Visual Studio можно дважды щелкнуть элемент управления в конструкторе, чтобы создать обработчик для его события по умолчанию: для элементов управления в виде кнопок это событие Click.

Реализация метода (который пока является пустым) находится в файле MainPage. xaml. cs. Вставьте следующий код в тело метода ClickMeButton_Click.

```
private void ClickMeButton_Click (object sender, RoutedEventArgs e)
{
BannerTextBlock. Text = MessageTextBox. Text;
MessageTextBox. Text = String. Empty;
}
```

Этот код считывает текст, введенный пользователем в текстовом поле, а затем создает баннер с этим текстом.

### *2.11.7 Проверка*

Проверим, работает ли приложение ожидаемым образом. Кроме того, мы зададим точку останова и с помощью отладчика пройдем по исходному коду, анализируя значения переменных, чтобы составить краткое представление о том, как с помощью Visual Studio выполнять отладку приложений, запущенных в эмуляторе.

При необходимости повторно откройте файл с выделенным кодом для страницы MainPage. xaml. Для этого щелкните данный файл правой кнопкой мыши в обозревателе решений и выберите команду View Code.

Теперь определите точку останова для прекращения выполнения в обработчике событий для кнопки **Click Me**. Для задания точки останова найдите первую строку метода **ClickMeButton_Click** исходного файла и щелкните в области серого поля, расположенного на левой стороне окна редактора рядом с этой строкой. Красный кружок указывает положение вставленной точки останова (рис.). Либо щелкните строку в окне редактора, чтобы выбрать ее, а затем нажмите клавишу F9.

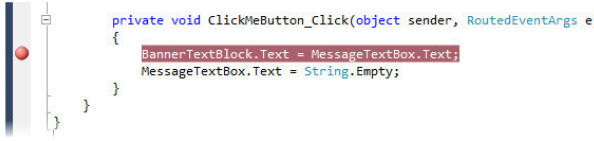



Рисунок 2.13 Установка точки останова

Чтобы включить или выключить точку останова, щелкните в области поля или щелкните строку, содержащую точку останова, и нажмите клавишу F9.

Для создания и развертывания приложения в эмуляторе Windows Phone нажмите клавишу F5 и начните сеанс отладки. Подождите, пока приложение запустится и появится его главная страница.

В окне эмулятора щелкните текстовое поле, чтобы активировать его. После этого появится экранная панель ввода (SIP). Введя тот или иной текст в текстовое поле, нажмите кнопку рядом с ним. Введенный текст должен отобразиться в верхней части программы.

Вернитесь в Visual Studio. Обратите внимание, что выполнение прекращается в заданной ранее точке останова и следующий выполняемый оператор выделяется желтым (рис.).

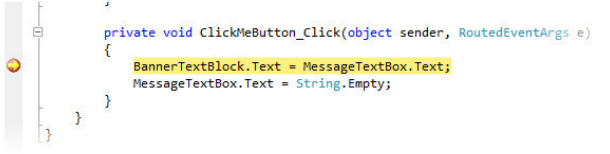



Рисунок 2.14 Срабатывание точки останова

Изучите текущее содержимое текстового поля в отладчике. Для этого в окне исходного кода наведите указатель мыши на свойство **MessageTextBox. Text**. Появится окно подсказки (совета) с текущим значением свойства, которое должно совпадать с текстом, введенным в окно эмулятора. Убедитесь, что указатель находится над частью Text. Иначе в подсказке будут отображаться сведения об объекте MessageTextBox (рис.).

alt=«Проверка значений переменных в отладчике»>

Нажмите клавишу F10, чтобы в пошаговом режиме выполнить текущую команду и отобразить в баннере текст, соответствующий содержимому текстового поля. Отобразите подсказку для свойства **BannerTextBlock. Text**, чтобы убедиться, что его значение соответствует значению текстового поля (рис.).

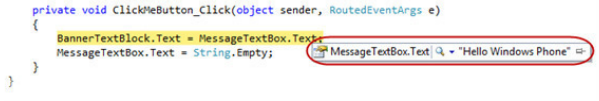



Рисунок 2.15 Проверка текущих значений переменной

При нажатии клавиши F10 отладчик выполняет текущую команду. А клавиша F11 обеспечивает пошаговое выполнение с заходом в вызываемые методы и функции. В этом случае, если команда включает вызов метода, отладчик выполняет заход в соответствующий метод для его отладки.

Нажмите клавишу F10 еще раз, чтобы выполнить следующий оператор и очистить содержимое текстового поля. Вновь отобразите совет для свойства **MessageTextBox. Text**, которое по-прежнему доступно, и убедитесь, что теперь оно

пустое. Нажмите клавишу F5, чтобы возобновить выполнение приложения. Вернитесь в эмулятор Windows Phone.

Нажмите в эмуляторе кнопку **Back**, чтобы перейти на предыдущую страницу. Обратите внимание, что при этом сеанс отладки завершается и в отладчике отображается главное меню: вы уходите с первой (и единственной) страницы приложения (закрываем ее), а других активных приложений нет.

Чтобы возобновить отладку после окончания текущего сеанса, нажмите клавишу F5 для повторного запуска приложения и присоединения отладчика. Однако обратите внимание, что при этом приложение запустится заново, а предыдущее состояние будет недоступно.

При закрытии эмулятора приложение останавливается, а отладчик отсоединяется. При отсоединении отладчик Visual Studio отображает сообщение о разрыве подключения к устройству.

***2.11. 8 Видео***

Вы также можете посмотреть видео о том, как создать первое приложение Windows Phone 7 – [2].

## 2.12 Создаем страницы с навигацией

Теперь мы узнаем, как можно сделать навигацию между страницами. В предыдущем примере приложение состояло всего лишь одной страницы. Иногда этого недостаточно.

В Silverlight навигация между страницами осуществляется очень просто. В этом вы сейчас сами убедитесь. Заодно вы увидите, как работает кнопка Back (Назад) в подобных приложениях. С ее помощью можно возвращаться на предыдущие страницы (напоминает поведение браузеров при посещении веб-страниц).

### *2.12.1 Создание приложения с навигацией*

Запустите Visual Studio и создайте новый проект **PageNavigation** (смотрите День первый). Далее необходимо добавить еще несколько новых страниц. Щелкните правой кнопкой мыши на имени проекта в Solution Explorer и выберите команду **Add→New Item…** и в диалоговом окне выберите элемент **Windows Phone Portrait Page** (портретная ориентация).

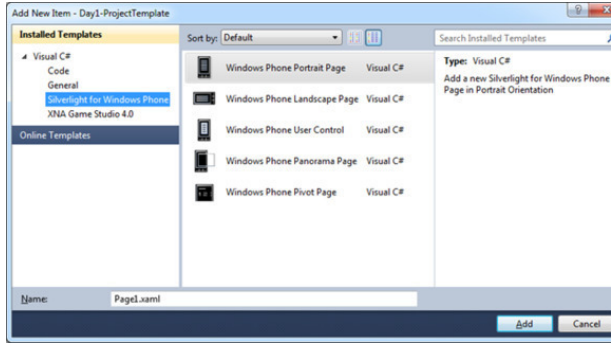



Рисунок 2.16 Выбор нового проекта

Сейчас мы пока не будем обсуждать вопросы, связанные с ориентацией экрана. Создайте три новых страницы Page1.xaml, Page2.xaml, Page3.xaml. Так как они выглядят совершенно одинаково, нам будет трудно ориентироваться среди них. Предлагаю сделать следующее. Откройте каждую созданную страницу и измените текст в них. Я, например, в место слов **PAGE NAME** использовал имена котов: Рыжик, Барсик, Васька. Теперь мы не запутаемся.

### *2.12.2 Создание гиперссылок на другие страницы*

Навигацию между страницами будем делать при помощи гиперссылок. Найдите на панели инструментов элемент HyperlinkButton и добавьте трижды данный элемент на панель эмулятора. После добавления измените код XAML следующим образом:

```
<HyperlinkButton Content=«Рыжик» NavigateUri="/page1.xaml»
Name=«hiperlinkbutton1» Height=«30» Width=«200»
HorizontalAlignment=«Left» VerticalAlignment=«Top»
Margin=«0,6,0,0» />
<HyperlinkButton Content=«Барсик» NavigateUri="/page2.xaml»
Name=«hiperlinkbutton2» Height=«30» Width=«200»
HorizontalAlignment=«Left» VerticalAlignment=«Top»
Margin=«0,6,0,0» />
<HyperlinkButton Content=«Васька» NavigateUri="/page3.xaml»
```

Name=«hiperlinkbutton3» Height=«30» Width=«200»
HorizontalAlignment=«Left» VerticalAlignment=«Top»
Margin=«0,6,0,0» />

Мы поменяли у гиперссылок текст, а также установили размеры и расположение на странице. При желании этого же результата можно добиться, изменяя соответствующие свойства в окне свойств. И самое главное – мы указали в атрибуте **NavigateUri** нужные имена страниц.

Удивительно, мы не написали еще ни одной строчки кода на C#, но тем не менее, приложение уже работает. Убедитесь сами. Запустите приложение и попробуйте нажимать на ссылки. Вы будете переходить на первую, вторую или третью страницу в зависимости от выбранной ссылки. Обратите внимание, что для возврата на основную страницу вы можете использовать аппаратную кнопку Back. При этом, если вы находитесь на главной странице и нажмете на кнопку Back, то тем самым вы закроете приложение.

### *2.12.3 Навигация через код*

Мы осуществили навигацию при помощи XAML-кода. Такого же результата можно добиться и через код на C#. Для этого добавим в проект три новых элемента Button (кнопка). Чтобы не писать одинаковый код для каждой кнопки, создадим общий обработчик событий для них. Для этого нам нужно знать имена кнопок.

**[XAML]**

<Button Content=«Рыжик» Height=«72»

```
HorizontalAlignment=«Left» Margin=«12,40,0,0»
   Name=«button1» VerticalAlignment=«Top» Width=«160»
   Click=«Button_Click» />
   <Button Content=«Барсик» Height=«72»
HorizontalAlignment=«Left» Margin=«12,120,0,0»
   Name=«button2» VerticalAlignment=«Top» Width=«160»
   Click=«Button_Click» />
   <Button Content=«Васька» Height=«72»
HorizontalAlignment=«Left» Margin=«12,200,0,0»
   Name=«button3» VerticalAlignment=«Top» Width=«160»
   Click=«Button_Click» />
```

**[C#]**

```
   private void Button_Click (object sender,
RoutedEventArgs e)
   {
   Button clickedbutton = sender as Button;
   switch (clickedbutton.Name)
   {
   case «button1»: NavigationService.Navigate (new Uri (»/
page1.xaml», UriKind.Relative));
   break;
   case «button2»:
   NavigationService.Navigate (new Uri (»/page2.xaml»,
UriKind.Relative));
   break;
```

```
case «button3»:
NavigationService.Navigate (new Uri (»/page3.xaml», UriKind.Relative));
break;
}
}
```

Снова запустите проект и убедитесь, что навигация при помощи кнопок работает так же, как и в примере с гиперссылками.

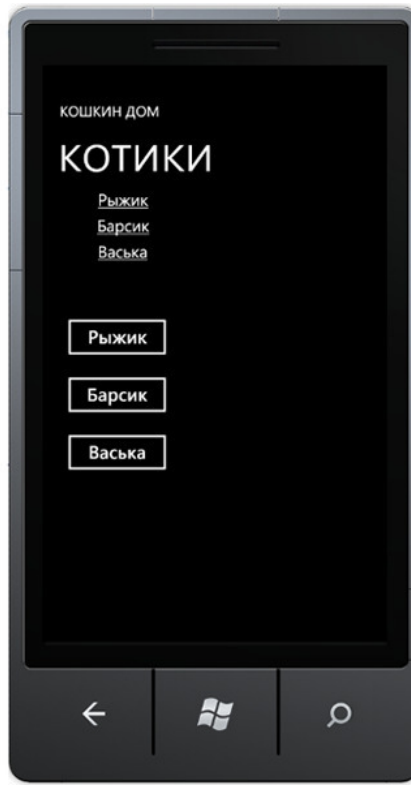



Рисунок 2.17 Вид программы в окне эмулятора

Для перехода назад вы можете использовать метод **NavigationService. GoBack**, который возвращает к экземпляру предыдущей страницы. Конечно, это дублирует функциональность кнопки «Назад», так что вы, скорее всего, бу-

дете вызывать этот метод как часть какой-либо другой функциональности.

```
private void button1_Click (object sender, RoutedEventArgs e)
{
NavigationService. GoBack ();
}
```

### *2.12.4 Передача параметров*

Иногда требуется не просто перейти на другую страницу, но и передать ей некоторые данные с предыдущей страницы. Добавьте на первую страницу текстовое поле и кнопку под именем **passParam**.

Добавьте код для обработчика щелчка кнопки

```
private void passParam_Click (object sender, RoutedEventArgs e)
{
NavigationService.Navigate (new Uri (»/SecondPage. xaml?msg=" + textBox1.Text, UriKind.Relative));
}
```

Вы видите, что при навигации на вторую страницу мы передаем строковые данные, которые берутся из текстового поля. Чтобы получить передаваемые данные, добавьте на второй странице текстовый блок (TextBlock) под именем textBlock1. В файле SecondPage. xaml. cs создайте следующий метод:

```
protected override void
```

```
OnNavigatedTo(System.Windows.Navigation.NavigationEventA
    {
    base. OnNavigatedTo (e);
    string msg = «»;
    if (NavigationContext.QueryString.TryGetValue («msg», out
msg)) textBlock1.Text = msg;
    }
```

Запустите приложение, введите текст на первой странице и убедитесь, что он отображается на второй странице.

### *2.12.5 Аппаратная кнопка Back*

Одним из требований к устройствам под управлением Windows Phone 7 является наличие аппаратной кнопки Back (Назад). По умолчанию кнопка Back работает как в обычном браузере. Она автоматически запоминает посещаемые страницы приложения, а также закрывает само приложение, если приложение состоит из одной страницы или пользователь вернулся к основной странице из других страниц и больше некуда возвращаться.

**Переопределяем работу кнопки Back**

Если вас не устраивает логика работы кнопки Back по умолчанию, то вы можете переопределить ее работу в нужном вам направлении. Делается это очень просто.

```
    protected override void
OnBackKeyPress(System.ComponentModel.CancelEventArgs e
    {
    // Ваш код здесь
```

```
e.Cancel = true; //Cancels the default behavior.
}
```

Давайте попробуем посмотреть работу переопределенной кнопки на примере. Возьмем предыдущий проект с навигацией по страницам PageNavigation и откроем редактор кода для страницы **Page3.xaml. cs**. Напишем следующий код:

```
protected override void OnBackKeyPress(System.ComponentModel.CancelEventArgs e
{
// Ваш код здесь
NavigationService.Navigate (new Uri (»/page1.xaml», UriKind.Relative));
e.Cancel = true; //Cancels the default behavior.
}
```

Запустим проект, нажмем на кнопку или ссылку **Васька**. Далее нажимаем аппаратную кнопку **Back** и убеждаемся, что теперь мы возвращаемся не на основную страницу MainPage. xaml, а на страницу Page1.xaml.

Как видите, код работает – мы переопределили работу аппаратной кнопки под свои нужды. Но не спешите радоваться. С переопределением кнопки Back нужно быть очень осторожным. Если вы еще не закрыли пример и находитесь на странице Page1.xaml (страница Рыжика), то нажмите на Back еще раз. Как и ожидалось, вы вернетесь обратно на Page3.xaml, с которого вы пришли. Нажимаем кнопку Back и замечаем, что снова попадаем на страницу Ры-

жика. Получился замкнутый круг – мы больше не можем никуда попасть и попеременно оказываемся на страницах Page1.xaml и Page3.xaml.

Поэтому тщательно тестируйте свою программу, если решили изменить поведение кнопки Back. Иначе, вы рискуете попасть в неприятную ситуацию.

**Кнопка Back для XNA-приложений**

Описанный нами пример в основном применяется для Silverlight-приложений. Мы еще не знакомы с разработкой приложений на основе XNA, но, забегая вперед, скажу, что в XNA-программах есть класс **GamePad**, имеющим свойство Buttons.Back. Поэтому вам придется встречаться с такой конструкцией:

```
// выйти, если нажата кнопка Back
if(GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
```

## 2.13 Ориентация дисплея

Пора поговорить о вашей ориентации. Если вы, разговаривая по телефону, держите его правой рукой, то вы… обычный пользователь, который держит телефон в портретной ориентации (вертикально). Если вы смотрите на устройстве фотографии, то удобнее его развернуть горизонтально – это уже альбомная ориентация. О том, как в программах для Windows Phone 7 управлять обеими ориентациями, и будет

посвящена сегодняшняя статья.

### *2.13.1 Ориентация*

Вам необходимо запомнить, что существуют два положения экрана. Вертикальное расположение называется портретным (Portrait), а горизонтальное – альбомным (Landscape). По умолчанию, приложения на Silverlight запускаются в портретном режиме, а XNA-приложения запускаются в альбомном, так как игры лучше смотрятся именно в этом режиме (посмотрите на ваш монитор). Но вы можете управлять режимами исходя из ваших задач. Жесткой привязки к ориентации нет. Более того, часто необходимо поддерживать два режима одновременно.

### *2.13.2 Рекомендация по проектированию интерфейса*

Если приложение поддерживает ввод текста, вы должны поддерживать альбомную ориентацию из-за возможности существования аппаратной клавиатуры.

Существуют различные способы гарантировать правильное отображение содержимого как в портретной, так и в альбомной ориентации. Два основных метода – это прокрутка (scrolling) и сетка (grid layout). Эти методы могут использоваться отдельно или в сочетании друг с другом.

Scrolling использует элемент управления StackPanel, который находится в элементе управления ScrollViewer. Используйте этот метод, если содержимое отображается в виде списка или если различные элементы управления следуют один за другим на странице. StackPanel позволяет установить

порядок расположения дочерних элементов одного за другим, а элемент управления ScrollViewer позволяет прокручивать содержимое StackPanel, если элементы пользовательского интерфейса не помещаются на экране.

### *2.13.3 Управление ориентацией экрана в Silverlight*

Если вы откроете предыдущие примеры на Silverlight и повернете устройство набок, то обнаружите, что изображение на экране не изменило свое расположение при смене ориентации. Это легко исправить. Посмотрите код в файле MainPage. xaml и найдите строчку:

SupportedOrientations=«Portrait» Orientation=«Portrait»

Данный код говорит о том, что проект поддерживает только портретный режим и в этом же режиме и запускается. Свойство **SupportedOrientations** поддерживает следующие режимы (рис.):

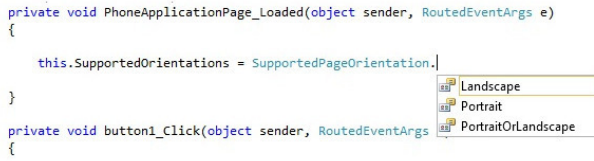



Рисунок 2.18 Набор свойств ориентации экрана

– Portrait (по умолчанию)
– Landscape
– PortraitOrLandscape

Достаточно заменить указанную выше строчку на **SupportedOrientations=«PortraitOrLandscape»**, и ваше приложение будет реагировать на смену режима автоматически.

Вероятно, вы уже догадались, что свойство Orientation отвечает за начальный режим во время запуска приложения. Только вы не должны забывать о поддержке выбранного режима. Иными словами, если вы хотите, чтобы приложение запускалось в альбомном режиме, то у свойства **SupportedOrientation** должно быть значение **Landscape** или **PortraitOrLandscape**. Возможно, вас удивит, что свойство поддерживает целых шесть различных значений.

– Landscape
– LandscapeLeft
– LandscapeRight
– Portrait
– PortraitDown
– PortraitUp

Используя эти значения, вы можете даже задать портретный режим вверх тормашками (PortraitUp). PortraitDown – это обычный портретный режим, а Portrait – общее свойство для двух портретных режимов, когда уточнение не требуется. Аналогично справедливо и для альбомных режимов.

### *2.13.4 События изменения ориентации*

Итак, вы можете настроить ориентацию экрана еще во время разработчки дизайна приложения через свойство SupportedOrientation. В этом случае система сама позаботится о нужном режиме. Тут нужно проявить аккуратность. Как правило, разработчик не препятствует пользователю вращать устройством во время работы с приложением. Но здесь может возникнуть одна небольшая неприятность. Взгляните на рисунки ниже. Предположим, вы написали программу-калькулятор, который красиво смотрится в портретном режиме. Вы ввели поддержку альбомного режима, и пользователь развернул телефон горизонтально. Как видите, нижняя часть интерфейса программы стала невидимой, уйдя за пределы экрана (рис.).

# Конец ознакомительного фрагмента.